#2

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005-3315

202 • 408 • 4000
FACSIMILE 202 • 408 • 4400

ATLANTA
404•653•6400
PALO ALTO
650•849•6600

WRITER'S DIRECT DIAL NUMBER:

(202) 408-4024

TOKYO
011•813•3431•6943
BRUSSELS
011•322•646•0353



September 22, 2000

ATTORNEY DOCKET NO.: 04329.2431

**Box Patent Application**
**Assistant Commissioner for Patents**
**Washington, D.C. 20231**

New U.S. Patent Application
Title: METHOD FOR DETERMINING A SERVER COMPUTER WHICH CARRIED OUT A PROCESS MOST RECENTLY, AND HIGH AVAILABILITY COMPUTER SYSTEM

Inventors: Kotaro ENDO and Koji YAMAMOTO

Sir:

We enclose the following papers for filing in the United States Patent and Trademark Office in connection with the above patent application.

1. A check for $730.00 representing the filing fee and $40.00 for recording the Assignment.

2. Application - 64 pages, including 3 independent claims and 15 claims total.

3. Drawings - 5 sheets of formal drawings containing 7 figures.

4. Declaration and Power of Attorney.

5. Recordation Form Cover Sheet and Assignment to Kabushiki Kaisha Toshiba.

6. Certified copy of Japanese Patent Application No. 11-364571, filed on December 22, 1999.

Applicants claim the right to priority based on Japanese Patent Application No. 11-364571, filed on December 22, 1999.

Please accord this application a serial number and filing date and record and return the Assignment to the undersigned.

The Commissioner is hereby authorized to charge any additional filing fees due and any other fees due under 37 C.F.R. § 1.16 or § 1.17 during the pendency of this application to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

By: _____
Richard V. Burgujian
Reg. No. 31,744

RVB/FPD/mld
Enclosures

# 日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: １９９９年１２月２２日

出 願 番 号
Application Number: 平成１１年特許願第３６４５７１号

出 願 人
Applicant (s): 株式会社東芝

CERTIFIED COPY OF
PRIORITY DOCUMENT

２０００年 ８月 ４日

特許庁長官
Commissioner,
Patent Office 及川耕造

出証番号 出証特２０００－３０６２２２３

【書類名】　特許願

【整理番号】　A009906608

【提出日】　平成11年12月22日

【あて先】　特許庁長官　殿

【国際特許分類】　G06F 15/16

G06F 17/00

H04L 29/14

【発明の名称】　最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体、及び高可用性計算機システム

【請求項の数】　8

【発明者】

【住所又は居所】　東京都府中市東芝町１番地　株式会社東芝府中工場内

【氏名】　山本　浩司

【発明者】

【住所又は居所】　東京都府中市東芝町１番地　株式会社東芝府中工場内

【氏名】　遠藤　浩太郎

【特許出願人】

【識別番号】　000003078

【氏名又は名称】　株式会社　東芝

【代理人】

【識別番号】　100058479

【弁理士】

【氏名又は名称】　鈴江　武彦

【電話番号】　03-3502-3181

【選任した代理人】

【識別番号】　100084618

【弁理士】

【氏名又は名称】　村松　貞男

【選任した代理人】

【識別番号】　100068814

【弁理士】

【氏名又は名称】　坪井　淳

【選任した代理人】

【識別番号】　100092196

【弁理士】

【氏名又は名称】　橋本　良郎

【選任した代理人】

【識別番号】　100091351

【弁理士】

【氏名又は名称】　河野　哲

【選任した代理人】

【識別番号】　100088683

【弁理士】

【氏名又は名称】　中村　誠

【選任した代理人】

【識別番号】　100070437

【弁理士】

【氏名又は名称】　河井　将次

【手数料の表示】

【予納台帳番号】　011567

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】　　明細書

【発明の名称】　　最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体、及び高可用性計算機システム

【特許請求の範囲】

【請求項１】　２台のサーバ計算機の一方が処理を行い、処理を行っているサーバ計算機で障害が発生した場合に、もう一方のサーバ計算機が処理を引き継ぐことが可能な高可用性計算機システムに適用され、前記各サーバ計算機に実行させるための最後に処理を行っていたサーバ計算機を判定するプログラムを記録した計算機読み取り可能な記録媒体であって、

前記各サーバ計算機に、

前記２台のサーバ計算機の少なくとも一方で障害が発生した場合と障害から復帰した場合とに、所定の状態遷移図に従って当該サーバ計算機の状態遷移を行う状態遷移ステップと、

前記状態遷移に応じて、少なくとも当該遷移後の自サーバ計算機の状態で決まるサーバ優先度を自サーバ計算機がローカルに持つ記憶装置に記録するサーバ優先度記録ステップと、

前記２台のサーバ計算機の両方に障害が発生し、その後少なくとも自サーバ計算機が障害から復帰した際に、少なくとも当該自サーバ計算機の前記記憶装置に記録されたサーバ優先度に基づいて当該自サーバ計算機の方が優先度が高いか否かを判定する優先度判定ステップと、

前記優先度判定ステップでの優先度判定結果に基づいて、自サーバ計算機が最後に処理を行っていたサーバ計算機であるか否かを判定することで、自サーバ計算機が処理を継続するか否かを決定し、その決定結果により前記状態遷移ステップでの状態遷移を行わせる最終処理計算機判定ステップと

を実行させるための最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体。

【請求項２】　前記優先度判定ステップは、前記２台のサーバ計算機の両方に障害が発生し、その後両サーバ計算機が共に障害から復帰した際に、当該両サーバ計算機の前記記憶装置に記録されたサーバ優先度を比較して、自サーバ計算

機の方が優先度が高いか否かを判定する第１の優先度判定ステップを含むことを特徴とする請求項１記載の最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体。

【請求項３】　前記優先度判定ステップは、前記２台のサーバ計算機の両方に障害が発生し、その後自サーバ計算機のみが障害から復帰した際に、自サーバ計算機の前記記憶装置に記録されたサーバ優先度が最高優先度であるか否かを判定する最高優先度判定ステップと、前記最高優先度判定ステップで最高優先度であると判定された場合にのみ自サーバ計算機の方が優先度が高いと判定する第２の優先度判定ステップを更に含むことを特徴とする請求項２記載の最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体。

【請求項４】　前記状態遷移ステップは、前記各サーバ計算機の状態を、処理を行い、且つ行っている処理を引き継ぐ相手が存在する「マスタ」と、処理は行うが、行っている処理を引き継ぐ相手が存在しない「シングルマスタ」と、処理を行っていないが、引き継ぎのための情報を受け取っている「スレーブ」と、処理を行っておらず、且つ引き継ぎのための情報を受け取っていない「停止」の４つの状態に分類して作成された前記状態遷移図に従って前記２台のサーバ計算機の状態遷移を行うことを特徴とする請求項３記載の最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体。

【請求項５】　前記状態遷移ステップは、前記２台のサーバ計算機の両方に障害が発生し、その後両サーバ計算機が共に障害から復帰した際に、前記最終処理計算機判定ステップでの判定結果と前記状態遷移図とに基づいて、前記２台のサーバ計算機の一方が前記停止状態から前記シングルマスタ状態となり、他方が前記停止状態から前記スレーブ状態となる状態遷移を行う第１の状態遷移ステップと、その後前記他方がスレーブ状態のまま、前記一方がマスタ状態となる状態遷移を行う第２の状態遷移ステップとを含むことを特徴とする請求項４記載の最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体。

【請求項６】　前記状態遷移ステップは、前記２台のサーバ計算機の両方に障害が発生し、その後自サーバ計算機のみが障害から復帰した際に、前記最終処理計算機判定ステップでの判定結果と前記状態遷移図とに基づいて、自サーバ計

算機が前記停止状態から前記シングルマスタ状態となる状態遷移を行うか、或いは現在の状態を維持する第３の状態遷移ステップと、その後相手サーバ計算機が障害から復帰した際に、当該相手サーバ計算機が前記停止状態から前記スレーブ状態となる状態遷移を行うか、或いは前記２台のサーバ計算機の一方が前記シングルマスタ状態となり、他方が前記スレーブ状態となる状態遷移を行う第４の状態遷移ステップとを含むことを特徴とする請求項４記載の最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体。

【請求項７】　前記状態遷移ステップは、前記２台のサーバ計算機の両方に障害が発生し、その後自サーバ計算機のみが障害から復帰した際に、当該自サーバ計算機の前記記憶装置に記録されたサーバ優先度が最高優先度でないために、前記最高優先度判定ステップでの判定ができず、当該自サーバ計算機が処理を継続する計算機であると判定できない状態で、当該自サーバ計算機に対して強制的に処理を継続させる強制開始命令が外部から与えられた場合に、当該自サーバ計算機が処理を継続するサーバ計算機として当該自サーバ計算機の状態遷移を行わせる強制開始ステップを含むことを特徴とする請求項３記載の最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体。

【請求項８】　２台のサーバ計算機の一方が処理を行い、処理を行っているサーバ計算機で障害が発生した場合に、もう一方のサーバ計算機が処理を引き継ぐことが可能な高可用性計算機システムにおいて、

前記各サーバ計算機は、

前記２台のサーバ計算機の少なくとも一方で障害が発生した場合と障害から復帰した場合とに、所定の状態遷移図に従って当該サーバ計算機の状態遷移を行う状態遷移手段と、

前記状態遷移手段により状態遷移が行われる度に、少なくとも当該遷移後の自サーバ計算機の状態で決まるサーバ優先度を自サーバ計算機がローカルに持つ記憶装置に記録する状態書き込み手段と、

前記２台のサーバ計算機の両方に障害が発生し、その後少なくとも自サーバ計算機が障害から復帰した際に、少なくとも当該自サーバ計算機の前記記憶装置に記録されたサーバ優先度に基づいて当該自サーバ計算機の方が優先度が高いか否

かを判定する優先度判定手段と、

前記優先度判定手段での優先度判定結果に基づいて、自サーバ計算機が最後に処理を行っていたサーバ計算機であるか否かを判定することで、自サーバ計算機が処理を継続するか否かを決定し、その決定結果により前記状態遷移手段による状態遷移を行わせる最終処理計算機判定手段と

を具備することを特徴とする計算機システム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、２台のサーバ計算機の一方が処理を行い、処理を行っているサーバ計算機で障害が発生しても、もう一方のサーバ計算機が処理を引き継ぐことができる高可用性計算機システムに係り、特に両方のサーバ計算機に障害が発生し、その後少なくとも一方が障害から復帰した場合に、その復帰した計算機が処理を継続するか否かを判定するのに必要な、最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体、及び高可用性計算機システムに関する。

【０００２】

【従来の技術】

従来から、複数のサーバ計算機（以下、サーバと略称する）、例えば２台のサーバをネットワーク等で結合し、１つのサーバで障害が発生しても、障害で停止した処理（サービス）を別のサーバが引き継ぐことにより、システム全体として可用性を維持できるようにしたクラスタ型の耐障害コンピュータシステム、つまり高可用性計算機システムが種々開発されている。

【０００３】

この種の計算機システムのうち、共有ディスク装置等の共有記憶装置を備えたシステムでは、一方のサーバで処理を実行しているときに、当該サーバから処理を引き継ぐのに必要な情報を当該共有記憶装置に記録しておくのが一般的である。

【０００４】

このような構成の計算機システムでは、例えば２台のサーバに障害が発生して、その後両サーバが共に障害から復帰した場合、或いは、いずれか一方のサーバだけが障害から復帰した場合のいずれの場合にも、共有記憶装置に記録された情報を利用することで、どの障害復帰サーバであっても容易に引き継ぎを行うことができる。

【０００５】

ところが、高可用性計算機システムの中には、共有記憶装置を持たないシステムもある。このような計算機システムでは、処理の引き継ぎを可能とするために、一方のサーバで処理を実行しているときに、当該サーバから他方のサーバに対し、処理の続きをするのに必要な情報を送るのが一般的である。

【０００６】

このようにすると、処理を実行しているサーバに障害が発生して処理を続行できなくなった場合に、他方のサーバでそれまでに送られてきた情報を使って処理の続きを行うこと、つまり処理を引き継ぐことが可能となる。

【０００７】

しかし、両方のサーバに障害が発生した場合に上記の処理の引き継ぎを行うのは容易ではない。その理由は、例えば両サーバが共に障害復帰（復旧）した場合であれば、いずれのサーバが処理を行う（引き継ぐ）かを決定しなければならないからである。また、いずれか一方のサーバだけが障害から復帰した場合であれば、その復帰したサーバで処理を行う（引き継ぐ）かを判定しなければならないからである。この従来技術について、以下に詳述する。

【０００８】

２台のサーバに障害が発生して、両サーバが共に障害復帰した場合、それまでに処理を行っていなかったサーバ（スレーブ）が処理の続きを行うためには、処理を行っていたサーバ（マスタ）から引き継ぎのための情報をもらっておく必要がある。

【０００９】

ところが、最後に処理を行っていたサーバ（マスタ）に障害が発生する直前に、このサーバからもう一方のサーバ（スレーブ）に対して、引き継ぎのための情

報を送っていなかった場合は、当該もう一方のサーバ（スレーブ）では処理の続きを行うことができない。そのため、両サーバが共に障害復帰した際には、最後に処理を行っていた方のサーバを、処理を引き継ぐサーバとして決定（選択）する必要がある。

【００１０】

次に、最後に処理を行っていたサーバ（マスタ）に障害が発生する前に、このサーバからもう一方のサーバ（スレーブ）に対して、引き継ぎのための情報を送っていた場合は、両サーバが共に障害復帰した際に、どちらのサーバを選択しても処理の続きをすることが可能である。しかし、障害発生の直前の処理については、最後に処理を行っていたサーバから引き継ぎに必要な情報をもう一台のサーバに送る前に障害が発生している可能性がある。そのため、こちらの場合も、最後に処理を行っていた方のサーバを選択する必要がある。

【００１１】

また、いずれか一方のサーバだけが障害から復帰した場合にも、そのサーバに処理をさせる条件は、両サーバが共に障害復帰した場合と同様の理由により、そのサーバが２台のうち最後まで処理を行っていたサーバであることである。

【００１２】

そのため従来は、最後に処理を行っていたサーバが判定可能なように、次に述べる２つの方法のいずれかを適用していた。

【００１３】

（１）引き継ぎを１度に限定する方法

予め、どちらか１つのサーバをプライマリサーバ、もう１つのサーバをセカンダリサーバと決めておき、初めは必ずプライマリサーバがマスタ、セカンダリサーバがスレーブとして動作を開始する。ここで、マスタは処理（クライアントから要求された処理）を行い、引き継ぎのための情報をスレーブに送る。スレーブはマスタから送られる引き継ぎのための情報を受け取って自身の有するディスク装置等の外部記憶装置（ローカルな外部記憶装置）に保存する。

【００１４】

そして、１度、プライマリサーバに障害が発生して引き継ぎを行ったなら、そ

の後はプライマリサーバが復帰しても利用しない。つまり、引き継ぎを１度に限定する。この場合、セカンダリサーバが、自身の外部記憶装置に自身が処理を行ったかどうかを記録しておくことにすれば、容易にどちらが最近まで処理を行っていたかを判定できる。

【００１５】

しかし、この従来技術は引き継ぎが１度しかできないため、片方または両方のサーバに障害発生、或いは障害復帰がいつ起こっても、可能な限り処理を継続するという、いわゆる自動運転が実現できない。

【００１６】

（２）時刻情報を使用する方法

予め、２台のサーバの時計（時刻））を合わせておく。サーバが処理を開始するときに、現在時刻を自身の持つ外部記憶装置に記録する。このようにすると、両サーバが共に障害復帰した際に、外部記憶装置に記録した時刻の情報をネットワークを通じて互いに授受し、より新しい時刻情報を持つサーバを、最後に処理を行っていたサーバとして判定する。

【００１７】

この時刻情報を使う方法は、時間というグローバルなものを暗黙に仮定しており、各サーバの使う時計が常に同期しているということを仮定している。しかし、実際の時計は必ずしも常に同期しているとは限らず、この方法は完全性に問題がある。また、一方のサーバだけが障害復帰した場合には、もう一方のサーバとの間で時刻情報の授受が行えないため、自身が最後に処理を行っていたサーバであるか否か判定できない。

【００１８】

【発明が解決しようとする課題】

上記したように、２台のサーバの一方が処理を行い、処理を行っているサーバで障害が発生しても、もう一方のサーバが処理を引き継ぐことが可能な高可用性計算機システムであって、共有記憶装置を持たない従来の高可用性計算機システムでは、２台のサーバに障害が発生し、その後少なくとも一方が障害から復帰した場合に、処理を継続するサーバ、即ち最後に処理を行っていたサーバを判定可

能とする方法として、「引き継ぎを１度に限定する方法」、或いは「時刻情報を使用する方法」が適用されていた。

【００１９】

しかし、「引き継ぎを１度に限定する方法」では、引き継ぎが１度しかできないため、自動運転が実現できないという問題があった。

また、「時刻情報を使用する方法」では、時間というグローバルなものを暗黙に仮定し、各サーバの使う時計が常に同期しているということを仮定しているが、実際の時計は必ずしも常に同期しているとは限らず、したがって判定精度（完全性）に問題があった。また、一方のサーバだけが障害復帰した場合には、自身が最後に処理を行っていたサーバであるか否か判定できないという問題もあった。

【００２０】

本発明は上記事情を考慮してなされたものでその目的は、時刻情報といったグローバルな情報を使わず、ローカルな情報だけを使って最後に処理を行っていたサーバを判定することにより、処理引き継ぎの回数の制限をなくして自動運転を実現すると共に、判定精度の向上が図れる、最後に処理を行っていたサーバ（サーバ計算機）を判定するプログラムを記録した記録媒体、及び高可用性計算機システムを提供することにある。

【００２１】

【課題を解決するための手段】

本発明の記録媒体は、２台のサーバの一方が処理を行い、処理を行っているサーバで障害が発生した場合に、もう一方のサーバが処理を引き継ぐことが可能な高可用性計算機システムに適用される、最後に処理を行っていたサーバを判定するプログラムであって、上記各サーバに、以下の各ステップ、即ち、上記２台のサーバの少なくとも一方で障害が発生した場合と障害から復帰した場合とに、所定の状態遷移図に従って当該サーバの状態遷移を行う状態遷移ステップと、この状態遷移に応じて、少なくとも当該遷移後の自サーバの状態で決まるサーバ優先度を自サーバがローカルに持つ記憶装置に記録するサーバ優先度記録ステップと、サーバの両方に障害が発生し、その後少なくとも自サーバが障害から復帰した

際に、少なくとも当該自サーバの記憶装置に記録されたサーバ優先度に基づいて当該自サーバの方が優先度が高いか否かを判定する優先度判定ステップと、この優先度判定ステップでの優先度判定結果に基づいて、自サーバが最後に処理を行っていたサーバであるか否かを判定することで、自サーバが処理を継続するか否かを決定し、その決定結果により上記状態遷移ステップでの状態遷移を行わせる最終処理計算機判定ステップとを実行させるための、最後に処理を行っていたサーバを判定するプログラムを記録したことを特徴とする。

【００２２】

このような構成においては、所定の状態遷移図に従う状態の遷移で決まるサーバ優先度という状態変数がローカルな記憶装置に記録され、当該状態遷移とローカルな状態変数を用いて、最後まで処理を行っていたサーバが判定されることから、処理引き継ぎの回数の制限をなくして自動運転を実現することが可能となる。

【００２３】

ここで、上記２台のサーバの両方に障害が発生し、その後両サーバが共に障害から復帰した際の処理のために、上記優先度判定ステップに次のステップ、即ち、上記両サーバが共に障害から復帰した際に、当該両サーバのサーバ優先度を比較して、自サーバの方が優先度が高いか否かを判定する第１の優先度判定ステップを持たせるとよい。

【００２４】

また、上記２台のサーバの両方に障害が発生し、その後自サーバのみが障害から復帰した際の処理のために、上記優先度判定ステップに次の２つのステップ、即ち、自サーバのサーバ優先度が最高優先度であるか否かを判定する最高優先度判定ステップと、ここで最高優先度であると判定された場合にのみ自サーバの方が優先度が高いと判定する第２の優先度判定ステップとを持たせるとよい。

【００２５】

また本発明は、上記状態遷移図により示される上記２台のサーバの状態として、処理を行い、且つ行っている処理を引き継ぐ相手が存在する「マスタ」と、処理は行うが、行っている処理を引き継ぐ相手が存在しない「シングルマスタ」と

、処理を行っていないが、引き継ぎのための情報を受け取っている「スレーブ」と、処理を行っておらず、且つ引き継ぎのための情報を受け取っていない「停止」の４つの状態の組み合わせを適用したことを特徴とする。

【００２６】

このように、処理を行うサーバを、行っている処理を引き継ぐ相手が存在する「マスタ」と、行っている処理を引き継ぐ相手が存在しない「シングルマスタ」とに分類して区別することで、両サーバのローカルな状態変数（サーバ優先度）が同一値となるのを防止して、当該ローカルな状態変数を用いて最後に処理を行っていたサーバを判定する場合の判定精度を向上することが可能となる。

【００２７】

この効果は、上記サーバ優先度記録ステップに次の４つのステップ、即ち、自サーバの状態がシングルマスタ状態に遷移する場合に、自サーバのサーバ優先度を最高優先度を示すように変更する第１のサーバ優先度記録ステップと、自サーバの状態がマスタ状態に遷移する場合に、自サーバのサーバ優先度を２番目の優先度を示すように変更する第２のサーバ優先度記録ステップと、自サーバの状態がスレーブ状態に遷移する場合に、自サーバのサーバ優先度を最低優先度を示すように変更する第３のサーバ優先度記録ステップと、自サーバの状態が停止状態に遷移する場合に、自サーバのサーバ優先度の変更を抑止する（元優先度を再度記録することと等価）サーバ優先度維持ステップとを持たせることにより、一層顕著となる。

【００２８】

上記４つのサーバ状態を適用した構成では、２台のサーバの両方に障害が発生し、その後両サーバが共に障害から復帰した際に、上記最終処理計算機判定ステップでの判定結果と状態遷移図とに基づいて、上記２台のサーバの一方が停止状態からシングルマスタ状態となり、他方が停止状態からスレーブ状態となる状態遷移を行い、その後上記他方がスレーブ状態のまま、上記一方がマスタ状態となる状態遷移を行うとよい。

【００２９】

また、２台のサーバの両方に障害が発生し、その後自サーバのみが障害から復

帰した際には、上記最終処理計算機判定ステップでの判定結果と状態遷移図とに基づいて、自サーバが停止状態からシングルマスタ状態となる状態遷移を行うか、或いは現在の状態を維持し、その後相手サーバが障害から復帰した際に、当該相手サーバが停止状態からスレーブ状態となる状態遷移を行うか、或いは一方がシングルマスタ状態となり、他方がスレーブ状態となる状態遷移を行うとよい。

【００３０】

また本発明は、上記状態遷移ステップに次のステップ、即ち、上記２台のサーバの両方に障害が発生し、その後自サーバのみが障害から復帰した際に、当該自サーバのサーバ優先度が最高優先度でないために、上記最高優先度判定ステップでの判定ができず、当該自サーバが処理を継続する計算機であると判定できない状態で、当該自サーバに対して強制的に処理を継続させる強制開始命令が外部から与えられた場合に、当該自サーバが処理を継続するサーバとして当該自サーバの状態遷移を行わせる強制開始ステップを持たせたことをも特徴とする。

【００３１】

このような構成においては、２台のサーバに障害が発生し、その後１台だけが障害復帰しても、その１台のサーバ優先度が最高優先度でないために、その１台の方が停止状態にあるもう１台より優先度が高いと判定することが不可能で、したがって、その１台が最後に処理を行っていたサーバであると判定できない場合でも、その１台に対して外部の計算機（コマンド送信用計算機）から強制開始命令を与えることで、強制的に処理を行わせることが可能となる。このように強制的に処理を開始させることができる機能は、処理の続きは無理でも、処理を再開した方が都合がよい場合に便利である。

【００３２】

ここで、停止状態にあるサーバの優先度が最高優先度のときに、障害復帰したサーバに対して強制的に処理を行わせると、両サーバが共に最高優先度となる。このような状態では、両サーバに共に障害が発生して、その後共に障害復帰した場合に、優先度の高いサーバを特定できず、したがって処理の引き継ぎを自動的に行うことはできないが、再度強制開始を行うことで容易に対処可能となる。

【００３３】

なお、上記した最後に処理を行っていたサーバを判定するプログラムを記録した記録媒体に係る本発明は、各計算機が当該プログラムで実現される機能手段を備えた高可用性計算機システムに係る発明としても成立し、当該プログラムで適用される手順を持つ方法に係る発明としても成立する。

【００３４】

【発明の実施の形態】

以下、本発明の実施の形態につき図面を参照して説明する。

図１は本発明の一実施形態に係る高可用性計算機システムの構成を示すブロック図である。

【００３５】

図１のシステムにおいて、ネットワーク５００には、２台のサーバ（サーバ計算機）１００ａ，１００ｂと、コマンド送信用計算機３００とが接続されている。またネットワーク５００には、サーバ１００ａまたは１００ｂからのサービスを受ける図示せぬクライアント（クライアント計算機）も接続されている。サーバ１００ａ，１００ｂには、それぞれ外部記憶装置としての（ハードディスク装置に代表される）ディスク装置２００ａ，２００ｂが接続されている。この計算機システムは共有記憶装置を持たない。

【００３６】

図１の計算機システムは、２台のサーバ１００ａ，１００ｂのうちの一方が処理を行い、処理を行っているサーバで障害が発生しても、もう一方のサーバが処理を引き継ぐことができるだけでなく、両サーバ１００ａ，１００ｂで共に障害が発生して、その後少なくとも一方が障害から復帰した場合にも、処理を引き継ぐサーバ、つまり最後に処理を行っていたサーバを正しく判定することが可能な仕組みを有していることに特徴がある。この仕組みについては後述する。

【００３７】

なお、本実施形態における「障害の発生」とは、ハードウェア障害、ソフトウェア障害などの他に、電源の切断などを含めて、処理の続行ができなくなる状態をいう。また、「障害からの復帰」とは、ハードウェア障害からの復帰、ソフトウェア障害からの復帰の他に、電源の投入などを含めて、処理の続行が行える状

態になることをいう。

【００３８】

さて、処理の引き継ぎを可能とするためには、単に同じ処理を別のサーバで行うだけではなく、それまで処理を行っていたサーバから、処理の続きを行うのに必要な情報を受け取っておかなければならない。ところが本実施形態のように共有記憶装置を持たないシステムでは、当該共有記憶装置を介してのサーバ間の情報授受ができないことから、処理を行うサーバが、引き継ぎに必要な情報を他方のサーバに送らなければならない。しかし、この方式では、障害の発生時期と引き継ぎに必要な情報の転送時期との関係で、［従来の技術］の欄で述べたように、最後に処理を行っていたサーバを判定することが困難となる場合がある。そこで、最後に処理を行っていたサーバが容易に判定できる方法として、［従来の技術］で述べた「引き継ぎを１度に限定する方法」、或いは「時刻情報を使用する方法」が知られているが、自動運転が実現できないとか判定精度（完全性）が不十分であるといった問題がある。

【００３９】

ここで自動運転について詳述する。

自動運転とは、［従来の技術］の欄で述べたように、片方または両方のサーバに障害発生、或いは障害復帰がいつ起こっても、可能な限り処理を継続する運転方法を指す。自動運転を実現するためには、少なくとも次の動作（１）～（５）が行える必要がある。

【００４０】

（１）２台のサーバに障害が発生し、その後２台のサーバが復帰したとき、処理を行うことが可能か否かを判定し、可能な場合は処理を行うべきサーバを決定し、処理を開始する。もう１台のサーバは、引継ぎ可能な状態にする。「引き継ぎ可能な状態にする」とは、処理を引き継ぐために必要な情報を、処理をしているサーバから、もう１台のサーバに全て送り、処理をしているサーバに障害が発生しても、もう１台のサーバが処理を続けられるようにすることを指す。

【００４１】

（２）２台のサーバに障害が発生し、その後１台のサーバが復帰したとき、復

帰したサーバで処理を行うことが可能か否かを判定し、可能な場合は処理を開始する。

【００４２】

（３）１台のサーバが処理中で、もう１台のサーバは障害が発生している状態のとき、障害が発生しているサーバが復帰した際に、そのサーバを引継ぎ可能な状態にする。

【００４３】

（４）１台のサーバが処理中で、もう１台のサーバが引継ぎ可能な状態にあるとき、処理中のサーバで障害が発生した場合は、引き継ぎを行う。

【００４４】

これに加え、次の動作（状態）は必ず考えなければならない。

（５）どのタイミングでも、障害が発生する。

【００４５】

これらの動作（状態）に基づき、図２に示す状態遷移図８００が作成（用意）される。ここでは、サーバの状態を、マスタ（マスタ状態）Ｍ、シングルマスタ（シングルマスタ状態）ＳＭ、スレーブ（スレーブ状態）ＳＬ、及び停止（停止状態）Ｘの４つの状態に分類している。つまり本実施形態では、従来のマスタＭを新たなマスタＭとシングルマスタＳＭという２状態に分類し、マスタＭ、スレーブＳＬ、及び停止Ｘの３つの状態にシングルマスタＳＭという状態が加えられた点に特徴がある。

【００４６】

サーバの状態は、まず、そのサーバが処理を実行しているか、実行していないかによって分けられ、

処理を行っている：

マスタＭ、シングルマスタＳＭ

処理を行っていない：

スレーブＳＬ、停止Ｘ

のようになる。

【００４７】

更に、「処理を行っている」「処理を行っていない」のそれぞれを、

処理を行っている：

行っている処理を引き継ぐ相手が存在する…マスタＭ

行っている処理を引き継ぐ相手が存在しない…シングルマスタＳＭ

処理を行っていない：

引き継ぎのための情報を受け取っている…スレーブＳＬ

引き継ぎのための情報を受け取っていない…停止Ｘ

のように分類する。

【００４８】

明らかなように、サーバに障害が発生した場合、処理を行わず、引き継ぎもできないので、当該サーバの状態は停止Ｘの状態になる。

【００４９】

さて、図２の状態遷移図８００では、一方のサーバ（例えばサーバ１００ａ）の状態をＡ、もう一方のサーバ（例えばサーバ１００ｂ）の状態をＢとすると、（Ａ　Ｂ）と表記している。（Ａ　Ｂ）と（Ｂ　Ａ）は異なる状態を表している。

【００５０】

図２の状態遷移図８００で適用されるそれぞれの状態について説明する。

まず（Ｘ　Ｘ）は、両サーバが共に停止状態にある状態を示す。具体的には、どちらのサーバも処理も行っておらず、引き継ぎもできない状態を示す。

【００５１】

（ＳＭ　ＳＬ）と（ＳＬ　ＳＭ）とは、一方のサーバがＳＭ（シングルマスタ）状態にあり、他方のサーバがＳＬ（スレーブ）状態にある状態を示す。具体的には、引き継ぎ可能な状態にするために、ＳＭの状態のサーバから、ＳＬの状態のサーバに、処理を引き継ぐための情報を送っている状態を示す。この状態では、一方のサーバで処理は行われているが、他方のサーバでの引き継ぎはできない。

【００５２】

（ＳＬ　Ｍ）と（Ｍ　ＳＬ）とは、一方のサーバがＭ（マスタ）状態にあり、

他方のサーバがＳＬ（スレーブ）状態にある状態を示す。具体的には、一方のサーバで処理が行われ、且つ他方のサーバで引き継ぐことができる状態である。

【００５３】

（Ｘ　ＳＭ）と（ＳＭ　Ｘ）とは、一方のサーバがＳＭ（シングルマスタ）状態にあり、他方のサーバがＸ（停止）状態にある状態を示す。具体的には、１台のサーバが処理を行っているが、もう１台のサーバに引き継ぎをするための情報を送ることができない状態である。つまり引き継ぎはできない。

【００５４】

次に、図２の状態遷移図８００を参照して、上記動作（１）～（５）について説明する。

動作（１）：

２台のサーバで共に障害が発生して停止している状態は、（Ｘ　Ｘ）という状態である。ここで、両サーバが障害から復帰すると、図２の１－１－１→１－１－２、または１－２－１→１－２－２の状態遷移が行われる。

【００５５】

このように、両サーバが障害から復帰した場合には、まず、どちらかのサーバで処理を始める。そして、引き継ぎ可能な状態にするために、引き継ぎに必要な情報を、処理を行っているサーバから、もう１台のサーバに送る状態に遷移する（１－１－１または１－２－１）。これが（ＳＬ　ＳＭ）または（ＳＭ　ＳＬ）状態である。

【００５６】

その後、引き継ぎが可能になったら、一方はスレーブ状態のまま、もう一方が（シングルマスタ状態から）マスタ状態という状態、つまり（ＳＬ　Ｍ）または（Ｍ　ＳＬ）の状態に移る（１－１－２または１－２－２）。この状態でも、引き継ぎに必要な情報はマスタからスレーブに送られる。

【００５７】

動作（２）：

両サーバが障害発生（＝停止）の状態（Ｘ　Ｘ）で、一方のサーバだけが障害から復帰した場合は、２－１または２－２の遷移が行われる。これは、復帰した

サーバを処理のできる状態（シングルマスタの状態）にするための状態遷移であり、（Ｘ　ＳＭ）または（ＳＭ　Ｘ）の状態に移る。

【００５８】

動作（３）：

一方のサーバが処理中で、他方のサーバが障害発生（＝停止）という状態は、（Ｘ　ＳＭ）または（ＳＭ　Ｘ）である。この（Ｘ　ＳＭ）または（ＳＭ　Ｘ）の状態で、停止状態のサーバが障害から復帰した場合、それぞれ３－１－１→３－１－２、または３－２－１→３－２－２の状態遷移が行われる。

【００５９】

このように、（Ｘ　ＳＭ）または（ＳＭ　Ｘ）の状態で停止状態のサーバが障害から復帰した場合には、まず、一方はシングルマスタ状態のまま、もう一方は（停止状態から）スレーブ状態という状態に移り（３－１－１または３－２－１）、引き継ぎに必要な情報を、処理を行っているサーバ（シングルマスタ）から、もう一方のサーバ（スレーブ）に送る。これが（ＳＬ　ＳＭ）または（ＳＭ　ＳＬ）状態である。

【００６０】

その後、引き継ぎが可能になったら、動作（１）と同様に、一方はスレーブ状態のまま、もう一方が（シングルマスタ状態から）マスタ状態という状態、つまり（ＳＬ　Ｍ）または（Ｍ　ＳＬ）の状態に移る（３－１－２または３－２－２）。これは、それまで処理を行っていたサーバをマスタ、復帰したサーバをスレーブとして、マスタ、スレーブを確立するための状態遷移である。

【００６１】

動作（４）：

引継ぎが可能な状態ということは、（ＳＬ　Ｍ）または（Ｍ　ＳＬ）という状態にあることである。この状態で、スレーブ状態のサーバに障害が発生した場合は、（Ｘ　ＳＭ）または（ＳＭ　Ｘ）という状態に遷移する（４－１または４－２）。これは、処理はそのまま同じサーバが継続するものの、引継ぎを行わない状態への状態遷移である。これに対し、マスタ状態のサーバに障害が発生した場合は、（ＳＬ　Ｍ）または（Ｍ　ＳＬ）の状態は（ＳＭ　Ｘ）または（Ｘ　ＳＭ

）の状態に遷移する（４－３または４－４）。これは、それまでスレーブ状態だったサーバが処理を引き継ぐための状態遷移である。

【００６２】

動作（５）：

サーバに障害が発生した場合、必ずそのサーバは停止状態（Ｘ）になってしまう。これが５－１乃至５－８である。以下にそれぞれの動作を説明する

まず、一方がマスタで、もう一方がスレーブの状態、つまり（ＳＬ　Ｍ）または（Ｍ　ＳＬ）の状態で、両方のサーバに障害が発生した場合には、５－１または５－５の遷移をして（Ｘ　Ｘ）の状態となる。

【００６３】

次に、シングルマスタ状態のサーバから引き継ぎを可能にするための情報をスレーブ状態のサーバに送っている（ＳＬ　ＳＭ）または（ＳＭ　ＳＬ）の状態で、シングルマスタ状態のサーバに障害が発生した場合には、引き継ぎはできないため、５－２または５－６の遷移が行われて、どちらのサーバも停止状態という状態、つまり（Ｘ　Ｘ）の状態となる。また、上記（ＳＬ　ＳＭ）または（ＳＭ　ＳＬ）の状態で両サーバに障害が発生した場合も、５－２または５－６の遷移が行われて、両サーバは停止状態になる。

【００６４】

また、上記（ＳＬ　ＳＭ）または（ＳＭ　ＳＬ）の状態で、スレーブ状態のサーバに障害が発生した場合は、５－３または５－７の遷移が行われて（Ｘ　ＳＭ）または（ＳＭ　Ｘ）の状態に移る。この状態では、処理は残ったサーバ（シングルマスタ状態のサーバ）で続行される。

【００６５】

次に、一方のサーバがシングルマスタ状態で、もう一方のサーバが停止状態、つまり（Ｘ　ＳＭ）または（ＳＭ　Ｘ）の状態で、シングルマスタ状態のサーバに障害が発生した場合には、５－４または５－８の遷移をして、両サーバ共に停止状態、つまり（Ｘ　Ｘ）となる。

【００６６】

以上のように、図２に示した状態遷移図８００には、自動運転を行うための動

作（１）～（５）が全て含まれている。

【００６７】

さて、上記の動作（１）～（５）のうち、動作（３）～（５）については、状態の変わるサーバがどちらであるかが決まっているため、制御することは容易である。

【００６８】

しかし、動作（１）は、処理を引き継ぐサーバが自身であるか否かを判定する必要があり、動作（２）も、障害から復帰したサーバがそのまま処理を引き継いでもよいか否かを判定する必要がある。本実施形態は、この動作（１）及び（２）において処理を引き継ぐサーバを判定する手法を、先に述べたサーバ状態の分類を利用して工夫した点に特徴がある。この判定手法の詳細については別途説明する。ここでは、処理を引き継ぎサーバは、最後まで処理を行っていた方のサーバとなることから、そのことについて述べる。

【００６９】

まず、動作（１）において、処理を引き継ぐべきサーバである条件は、２台のサーバのうち、最後まで処理を行っていた方のサーバであることである。その理由を以下に説明する。

【００７０】

本実施形態において、それまでに処理を行っていなかったサーバが処理の続きを行うためには、処理を行っていたサーバから、引き継ぎのための情報を受け取っておく必要がある。もし、最後に処理を行っていたサーバに障害が発生する直前に、当該サーバからもう一方のサーバに引き継ぎのための情報を送っていなかった場合は、もう一方のサーバでは処理の続きを行うことができない。そのため、最後に処理を行っていた方のサーバを、処理を引き継ぐべきサーバとする必要がある。

【００７１】

次に、最後に処理を行っていたサーバに障害が発生する前に、当該サーバからもう一方のサーバに引き継ぎのための情報を送っていた場合は、どちらのサーバでも処理の続きをすることが可能なように見える。しかし、障害発生の直前に行

っていた処理については、当該処理の引き継ぎに必要な情報を上記もう一方のサーバに送る前に、障害が発生している可能性がある。そのため、こちらの場合も、最後に処理を行っていた方のサーバを、処理を引き継ぐべきサーバとする必要がある。

【００７２】

以上から、動作（１）では、最後まで処理を行っていたサーバが処理を引き継ぐ必要がある。

【００７３】

次に動作（２）において、障害から復帰したサーバに処理をさせる条件は、動作（１）の場合と同じ理由により、当該障害復帰サーバが、２台のうち最後まで処理を行っていた方のサーバであることである。

【００７４】

さて、各サーバにおいて、自身が最後まで処理を行っていたサーバであるか否かを判定する手法として、［従来の技術］の欄で述べたような「時刻情報を使う方法」がある。しかし、「時刻情報を使う方法」は、時間というグローバルなものを暗黙に仮定し、各サーバの使う時計が常に同期しているという仮定のもとで成立しており、実際の時計は必ずしも常に同期しているとは限らないことから、完全性に問題がある。

【００７５】

このため本実施形態では、最後に処理を行っていたサーバをローカルな情報だけを使って判定する手法を適用している。但し、ローカルな情報には、次のような問題がある。それは、ローカルな情報は、当該情報を持つサーバが停止してしまうと、他のサーバから変更できないため、それぞれのサーバの持つ情報が矛盾してしまう可能性があるという問題である。

【００７６】

この問題について、処理を行っているサーバが、自分で処理を行っている旨の情報をローカルに持つ場合を例に述べる。この例では、処理をしているサーバは「自分が処理中」という情報を持っている。その後、このサーバに障害が発生し、相手サーバが処理を引き継いだ場合、もう一方のサーバも「自分が処理中」と

いう情報を持つことになる。このとき、障害が発生したサーバの持つ「自分が処理中」という情報を書き換えることはできない。このような状態で、今度は、もう一方のサーバにも障害が発生し、その後、両方のサーバが障害から復帰したとする。このとき、両方のサーバが「自分が処理中」という情報を持つことになる。したがって、処理をしているサーバが「自分が処理中」という情報をローカルに持つだけでは、どちらが最後に処理をしていたかを判定できない。

【００７７】

そこで本実施形態では、シングルマスタ状態を含む４つの状態に分類される各サーバの状態を、図２に示した状態遷移図８００に基づいて制御し、後述するサーバ優先度と称する３つの状態をとり得る状態変数をローカルな情報として持つことにより、この問題を解決するようにしている。

【００７８】

ここで、再び図１を参照すると、サーバ１００ａ，１００ｂは同一構成を有している。即ちサーバ１００ａ，１００ｂは、いずれもクラスタ管理部１１０、処理・引き継ぎ部４００、及びサーバ優先度処理部７００を備えている。これら各部１１０，４００，７００は、サーバ１００ａ，１００ｂが所定のソフトウェアプログラムを読み込み実行することにより実現される機能手段である。ここでは、クラスタ管理部１１０を実現するためのソフトウェア（クラスタソフトウェア）と、処理・引き継ぎ部４００を実現するためのソフトウェア（処理ソフトウェア）と、サーバ優先度処理部７００を実現するためのサーバ優先度管理ソフトウェアとは、同一の記録媒体、例えばＣＤ－ＲＯＭに記録して提供され、サーバ１００ａ，１００ｂの持つディスク装置２００ａ，２００ｂにインストールして用いられる。なお、上記各ソフトウェアがディスク装置２００ａ，２００ｂに予めインストールされていてもよく、また、それぞれ別々の記録媒体に記録されていてもよい。また、ネットワーク５００を介してダウンロードされるものであっても構わない。

【００７９】

クラスタ管理部１１０は、図２に示した状態遷移図８００に従う状態遷移制御を行う機能を有する。即ちクラスタ管理部１１０は、もう一方の（サーバ上で動

作する）クラスタ管理部１１０とネットワーク５００を通じて通信することにより、予め定められた状態遷移図８００に基づき、２台のサーバ１００ａ，１００ｂの状態遷移制御を行う。またクラスタ管理部１１０は、状態遷移の際には、処理・引き継ぎ部４００とサーバ優先度処理部７００とに、最新の遷移状態を示す状態変化情報９０１を送る。

【００８０】

またクラスタ管理部１１０は、サーバ１００ａ及び１００ｂの両サーバが停止状態（Ｘ）にある状態で、自身が動作するサーバ（自サーバ）が障害から復帰した際には、サーバ優先度処理部７００に自サーバの障害復帰を示す状態変化情報９０１を送る。更にクラスタ管理部１１０は、相手サーバのクラスタ管理部１１０との通信により、相手サーバの障害発生と、障害からの復帰とを検出し、状態遷移を行う。なおクラスタ管理部１１０は、自サーバの障害復帰時には、復帰後所定時間を待って、相手サーバも障害から復帰したか否かを当該相手サーバとの通信により調べ、しかる後に状態変化情報９０１を送るようにしている。

【００８１】

処理・引き継ぎ部４００は、クラスタ管理部１１０から得られる（最新の遷移状態を示す）状態変化情報９０１に応じて、処理と処理引き継ぎのための動作の開始／停止の制御を行う。処理・引き継ぎ部４００は、処理制御部４１０と、引き継ぎ制御部４２０とから構成される。

【００８２】

処理制御部４１０は処理の開始／停止を制御し、引き継ぎ制御部４２０は、処理の引き継ぎのための動作の開始／停止を制御する。処理とは、例えばＤＢＭＳ（データベースマネージメントシステム）等のアプリケーションプログラムの実行である。

【００８３】

サーバ優先度処理部７００は、自サーバ１００ｉ（ｉはａまたはｂ）のディスク装置２００ｉ（に確保された優先度記録領域）に自サーバ１００ｉの優先度（サーバ優先度）２１０を記録し、当該優先度２１０を相手サーバ１００ｊ（ｊはａまたはｂ、但しｉ≠ｊ）のディスク装置２００ｊに記録された優先度２１０と

比較する。このサーバ優先度２１０は最後まで処理を行っていたサーバを判定するために必要な情報であり、１，２，３の３状態のいずれかをとる。

【００８４】

サーバ優先度処理部７００は、状態書き込み部７１０と、比較処理部７２０とから構成される。

状態書き込み部７１０は、クラスタ管理部１１０から得られる状態変化情報９０１に応じて、自サーバ１００ｉのディスク装置２００ｉ（に確保された優先度記録領域）に記録されている自サーバ１００ｉの優先度２１０を更新する。

【００８５】

比較処理部７２０は、クラスタ管理部１１０から自サーバの障害復帰を示す状態変化情報９０１を受け取った場合、相手サーバの比較処理部７２０との間でネットワーク５００を介して優先度２１０の授受を行う。そして比較処理部７２０は、自サーバ１００ｉの優先度２１０と相手サーバ１００ｊの優先度２１０とを比較することで、自サーバ１００ｉの方が優先度が高いか否かの判定を行う。この判定の詳細は後述する。比較処理部７２０での優先度比較・判定結果は、クラスタ管理部１１０で行われる、自サーバ１００ｉが最後に処理をしたサーバであるか否か（つまり自サーバ１００ｉがマスタとなるサーバであるか否か）の判定に用いられる。

【００８６】

コマンド送信用計算機３００は、ユーザから要求された強制開始命令を、ネットワーク５００を介して、サーバ１００ａ，１００ｂのうち当該命令の指定するサーバのクラスタ管理部１１０に送信する。この強制開始命令は、指定のサーバで強制的に処理を開始させる命令である。

【００８７】

ここで、強制開始命令を用いた処理の強制開始機能を導入する技術的背景について述べる。

サーバ１００ａ及び１００ｂの両サーバに障害が発生し、その後、一方のサーバ１００ｉだけが障害から復帰した場合、つまり上記動作（２）の場合、その障害復帰サーバ１００ｉは処理を行うべきサーバではないと判定されることがあり

得る。この場合の動作としては、もう一方の、本来処理を行うべきサーバが復帰するまで待つことが考えられる。

【００８８】

しかし、もう一方のサーバが復帰するまで処理が停止したままであるよりは、処理の続きができなくても構わないので、先に障害から復帰したサーバ１００ｉで処理を再開した方が都合がよい場合も当然あり得る。

【００８９】

そこで本実施形態では、このような場合に、先に障害復帰したサーバで強制的に処理を開始できる強制開始モードと、本来処理を行うべきサーバが障害から復帰するまで待つ待機モードの２種のモードの中から、いずれか一方のモードが選択指定できるようにしている。ここでは、通常は待機モードに自動設定され、コマンド送信用計算機３００から強制開始命令が与えられた場合だけ、強制開始モードに切り替えられる構成を適用している。

【００９０】

次に、サーバ１００ａ，１００ｂの動作の詳細について、サーバ優先度処理部７００の動作を中心に説明する。

処理・引き継ぎ部４００は、クラスタ管理部１１０から最新の遷移状態を示す状態変化情報９０１が送られると、当該状態変化情報９０１を受け取り、当該状態変化情報９０１の示す最新の状態に従って、以下に述べる、処理の開始もしくは停止、または引き継ぎのための動作の開始もしくは停止を行う。

【００９１】

まず、処理の開始、停止については、前述したマスタ、シングルマスタ、スレーブ、停止の各状態の説明に従い、処理・引き継ぎ部４００内の処理制御部４１０が次のように動作する。

【００９２】

自サーバ１００ｉがマスタ（Ｍ）、またはシングルマスタ（ＳＭ）の状態となったときは処理を行う。

これに対し、自サーバ１００ｉがスレーブ（ＳＬ）、または停止状態（Ｘ）となったときは処理を行わない。

【００９３】

次に、引き継ぎのための動作の開始、停止については、処理・引き継ぎ部４００内の引き継ぎ制御部４２０が次のように動作する。

一方がシングルマスタまたはマスタ、もう一方がスレーブの状態になった状態では、処理引き継ぎのための情報の授受をネットワーク５００を介して行う。ここでは、シングルマスタまたはマスタ側の引き継ぎ制御部４２０が引き継ぎのための情報の送り手となり、スレーブ側の引き継ぎ制御部４２０が当該情報の受け手となる。

他の状態の組み合わせでは、引き継ぎのための情報授受は行わない。

【００９４】

次に、サーバ優先度処理部７００の動作の詳細について、サーバ優先度処理部７００を構成する状態書き込み部７１０と比較処理部７２０のそれぞれの動作に分けて順に説明する。

まず、サーバ優先度処理部７００内の状態書き込み部７１０は、クラスタ管理部１１０から最新の遷移状態を示す状態変化情報９０１が送られた場合、当該状態変化情報９０１の示す最新の状態に従って、自サーバ１００ｉのディスク装置２００ｉ（に確保された優先度記録領域）に記録されているサーバ優先度２１０を変更するサーバ優先度変更（記録）処理を、図３のフローチャートに従って次のように行う。

【００９５】

自サーバ１００ｉの状態がシングルマスタ状態（ＳＭ）に遷移したときは、１に変更する（ステップＳ１，Ｓ２）。

自サーバ１００ｉの状態がマスタ状態（Ｍ）に遷移したときは、２に変更する（ステップＳ１，Ｓ３）。

自サーバ１００ｉの状態がスレーブ状態（ＳＬ）に遷移したときは、３に変更する（ステップＳ１，Ｓ４）。

自サーバ１００ｉの状態が停止状態（Ｘ）に遷移したときは、変更しない（ステップＳ１，Ｓ５）。

【００９６】

ここで、サーバ優先度２１０の初期値は、サーバ１００ａと１００ｂとで異なり、一方のサーバでは２に、もう一方のサーバでは３に設定される。このサーバ優先度２１０の初期設定は、自動運転を始める前の処理で行われる。

【００９７】

次に、サーバ優先度処理部７００内の比較処理部７２０の動作、つまり最後に処理を行っていたサーバをクラスタ管理部１１０にて判定するのに必要な優先度比較・判定動作について、強制開始を行わない場合と行う場合とに分けて順次説明する。

【００９８】

まず、強制開始を行わない場合の動作を、図４のフローチャートを参照して説明する。

図２に示した状態遷移図８００において、最後に処理を行っていたサーバを判定する必要があるのは、１－１－１→１－１－２または１－２－１→１－２－２の遷移の場合と、２－１または２－２の遷移の場合の２つ、つまり２台のサーバ１００ａ，１００ｂに障害が発生して共に停止状態（Ｘ　Ｘ）となっているときに、両サーバが障害から復帰した場合と、いずれか一方のサーバが障害から復帰した場合である。なお、以下の説明では、１－１－１→１－１－２，１－２－１→１－２－２を、その共通部分をとって１－１，１－２で表現する。

サーバ１００ａ，１００ｂのクラスタ管理部１１０は、（Ｘ　Ｘ）の状態で自サーバが障害復帰すると、その旨を示す状態変化情報９０１を自サーバのサーバ優先度処理部７００に送る。

【００９９】

サーバ優先度処理部７００内の比較処理部７２０は、自サーバのクラスタ管理部１１０から自サーバが障害復帰したことを示す状態変化情報９０１が送られた場合、上記１－１もしくは１－２の遷移、または２－１もしくは２－２の遷移のために、自サーバが最後に処理を行っていたか否かの判定（自サーバが処理を引き継ぐか否かの判定）に必要なサーバ優先度２１０の比較・判定が要求されたものと判断し、当該比較・判定を次のように行う。

【０１００】

１－１もしくは１－２の遷移となる場合：

（１）比較処理部７２０は、相手サーバ（サーバ１００ｊとする）の比較処理部７２０とネットワーク５００を介して通信を行ってサーバ優先度２１０を授受し、自サーバ（サーバ１００ｉとする）と相手サーバ１００ｊとで、どちらのサーバ優先度２１０が高いかを比較・判定する（ステップＳ１１，Ｓ１２）。ここでサーバ優先度２１０は１が最も高く、２，３の順に低くなる。つまり、サーバ優先度２１０は１で最高優先度を、２で２番目の優先度を、３で最低優先度を示す。なお、自サーバ１００ｉのサーバ優先度２１０が１の場合には、相手サーバ１００ｊのサーバ優先度２１０を取得することなく、自サーバ１００ｉの方が優先度が高いと判定することも可能である。この理由については後述する。

【０１０１】

（２）比較処理部７２０は、サーバ優先度２１０の比較・判定結果、つまり自サーバ１００ｉの方が優先度が高いか否かのサーバ優先度判定結果を、自サーバのクラスタ管理部１１０から（Ｘ　Ｘ）状態で送られた状態変化情報９０１に対する応答として、当該クラスタ管理部１１０に通知する（ステップＳ１３）。

【０１０２】

（３）クラスタ管理部１１０は、比較処理部７２０からサーバ優先度判定結果を受け取ると、当該判定結果と相手サーバ１００ｊも障害復帰しているか否かと状態遷移図８００とに基づき、図５のフローチャートに従って次のように状態遷移する。

【０１０３】

まずクラスタ管理部１１０は、比較処理部７２０からのサーバ優先度判定結果により自サーバ１００ｉの方がサーバ優先度２１０が高いと判定できる場合（ステップＳ２１，Ｓ２２）、自サーバ１００ｉが最後に処理を行っていたサーバであると判断する。しかも、相手サーバ１００ｊも障害復帰している場合（ステップＳ２３）、クラスタ管理部１１０は、最終的に自サーバ１００ｉがマスタとなるために、まずシングルマスタ状態（ＳＭ）となり、相手サーバ１００ｊがスレーブ状態（ＳＬ）となるように遷移を行う（ステップＳ２４）。これに対し、相手サーバ１００ｊの方がサーバ優先度２１０が高い場合には（ステップＳ２５）

、自サーバ１００ｉは最後に処理を行っていたサーバではなく、最終的に相手サーバ１００ｊがマスタとなるものとして、自サーバ１００ｉがスレーブ（ＳＬ）状態となり、相手マスタ１００ｊがシングルマスタ状態（ＳＭ）となるように遷移を行う（ステップＳ２６）。つまり、図２の１－１－１（ＳＬ　ＳＭ）または１－２－１（ＳＭ　ＳＬ）の状態遷移が行われる。なお、相手サーバ１００ｊが障害復帰していない場合には（ステップＳ２３）、後述する２－１（Ｘ　ＳＭ）もしくは２－２（ＳＭ　Ｘ）の遷移となる（ステップＳ２７）。

【０１０４】

（４）クラスタ管理部１１０は、この状態遷移結果、つまり最新の遷移状態（ＳＬ　ＳＭ）または（ＳＭ　ＳＬ）を示す状態変化情報９０１を処理・引き継ぎ部４００及びサーバ優先度処理部７００に送る。この際の処理・引き継ぎ部４００（内の処理制御部４１０及び引き継ぎ制御部４２０）と、サーバ優先度処理部７００内の状態書き込み部７１０の動作内容は、先の説明から明らかであり、（サーバ１００ｉまたは１００ｊに再び障害が発生しないならば）最終的にはサーバ優先度２１０の高い方のサーバがマスタとなるように遷移する。つまり、図２の１－１－２（ＳＬ　Ｍ）または１－２－２（Ｍ　ＳＬ）の状態に遷移する。

【０１０５】

２－１もしくは２－２の遷移となる場合：

（１）比較処理部７２０は、相手サーバ１００ｊの比較処理部７２０とネットワーク５００を介して通信を行ってサーバ優先度２１０を授受し、自サーバ１００ｉと相手サーバ１００ｊとで、どちらのサーバ優先度２１０が高いかを比較する。ところが、２－１もしくは２－２の遷移となる場合には、相手サーバ１００ｊはまだ停止状態（Ｘ）にあることから、相手サーバ１００ｊのサーバ優先度２１０を取得することができない。しかし、自サーバ１００ｉのサーバ優先度２１０が１であれば、後述するように自サーバ１００ｉの方が優先度が高いと判断できる。

【０１０６】

（２）そこで比較処理部７２０は、相手サーバ１００ｊが停止状態にあるために当該相手サーバ１００ｊのサーバ優先度２１０が取得できない場合には（ステ

ップＳ１１）、自サーバ１００ｉのサーバ優先度２１０が１（つまり最高優先度）であるか否かを判定する（ステップＳ１４）。もし、自サーバ１００ｉのサーバ優先度２１０が１の場合には、比較処理部７２０は相手サーバ１００ｊのサーバ優先度２１０との比較を行うことなく、自サーバ１００ｉの方がサーバ優先度２１０が高いことを示すサーバ優先度判定結果を、自サーバ１００ｉのクラスタ管理部１１０に通知する（ステップＳ１５，Ｓ１３）。

【０１０７】

これに対し、自サーバ１００ｉのサーバ優先度２１０が１でない場合には（ステップＳ１４）、自サーバ１００ｉのサーバ優先度２１０だけでは、自サーバの方が優先度が高いか否かを判定できないとして、判定不可（不定）を示す優先度判定結果を自サーバ１００ｉのクラスタ管理部１１０に通知する（ステップＳ１６，Ｓ１３）。なお、自サーバ１００ｉのサーバ優先度２１０が３の場合には、自サーバ１００ｉの方が優先度が低い、つまり相手サーバ１００ｊの方が優先度が高いと判定できる。しかし、この例のように相手サーバ１００ｊが停止状態にある場合には状態遷移できない。そこで本実施形態では、相手サーバ１００ｊが障害復帰していない場合、自サーバ１００ｉのサーバ優先度２１０が１でない限り、全て判定不可として扱うようにしている。

【０１０８】

（３）クラスタ管理部１１０は、比較処理部７２０からサーバ優先度判定結果を受け取ると、当該判定結果と相手サーバ１００ｊも障害復帰しているか否かと状態遷移図８００とに基づき、図５のフローチャートに従って次のように状態遷移する。

【０１０９】

まずクラスタ管理部１１０は、相手サーバ１００ｊが障害復帰していない状態で、自サーバ１００ｉの方がサーバ優先度２１０が高い場合には（ステップＳ２１～Ｓ２３）、自サーバ１００ｉが最後に処理を行っていたサーバであるとして、状態遷移図８００に従い、自サーバ１００ｉがシングルマスタになるように遷移を行う（ステップＳ２７）。これに対し、判定結果が判定不可を示す場合（相手サーバ１００ｊが障害復帰しておらず、且つ自サーバ１００ｉのサーバ優先度

２１０が１でない場合）には（ステップＳ２５）、相手サーバ１００ｊが障害復帰するまで状態遷移を行わない。

【０１１０】

図２に示した状態遷移図８００に従い、上記の判定に基づいてサーバ状態が遷移したとすると、サーバ優先度２１０は図６のように遷移する。この図では、一方のサーバの状態をＡ、サーバ優先度をａ、もう一方のサーバの状態をＢ、サーバ優先度をｂとしたとき、（Ａ　Ｂ）と表記し、その下に［ａ　ｂ］と表記している。（Ａ　Ｂ）と（Ｂ　Ａ）は異なる状態を表す。［ａ　ｂ］と［ｂ　ａ］は異なる場合を表す。

【０１１１】

以下で、上述の判定法によって、最後に処理を行っていたサーバを正しく判定できる理由について説明する。

命題１　サーバ優先度２１０の高いサーバが最後に処理を行っていたサーバである。

命題２　あるサーバのサーバ優先度２１０が１であれば、そのサーバは必ず最後に処理を行っていたサーバである。

【０１１２】

この２つの命題１，２を証明する。そのためには、次の３つの補題１～３が成り立てばよい。

【０１１３】

補題１

仮定：

・ある時点で、自サーバ１００ｉのサーバ優先度２１０に１がセットされている。その後、自サーバ１００ｉのサーバ優先度２１０は変更されておらず、相手サーバ１００ｊがマスタ状態またはシングルマスタ状態になっていない。

・相手サーバ１００ｊのサーバ優先度２１０が１ではない。

結論：

自サーバ１００ｉのサーバ優先度２１０が１であるとき、最後に処理を行ったのは自サーバ１００ｉである。

【０１１４】

補題２

仮定：

・ある時点で、自サーバ１００ｉのサーバ優先度２１０に２がセットされている。その後、自サーバ１００ｉのサーバ優先度２１０は変更されておらず、相手サーバ１００ｊがマスタ状態またはシングルマスタ状態になっていない。

・相手サーバ１００ｊのサーバ優先度２１０が３である。

結論：

自サーバ１００ｉのサーバ優先度２１０は２であり、相手サーバ１００ｊのサーバ優先度２１０が３であるとき、最後に処理を行ったのは自サーバ１００ｉである。

【０１１５】

補題３

仮定：

両サーバ１００ｉ，１００ｊが停止した状態である。

結論：

サーバ優先度２１０は、互いに異なる値をとる。

【０１１６】

命題１の証明：

まず、補題１，２，３の結論を使い、命題１を証明する。

命題１を証明するには、サーバ優先度２１０の全ての組み合わせにおいて、優先度の高いサーバが最後に処理を行っていたことを証明できればよい。

【０１１７】

補題３から、両サーバ１００ｉ，１００ｊが停止した状態のとき、サーバ優先度２１０は互いに異なっているから、（１，２，３の３状態を持つ）サーバ優先度２１０の組み合わせは、（１，２）（１，３）（２，３）の３通りである。

【０１１８】

補題１から、（１，２）（１，３）についてはサーバ優先度２１０が１のサーバが最後に処理を行っている。次に補題２から、（２，３）については、サーバ

優先度２１０が２のサーバが最後に処理を行っている。

よって、全ての場合において、サーバ優先度２１０の高いサーバが最後に処理を行っているといえる。

【０１１９】

命題２の証明：

次に、補題１，３の結論を使って命題２を証明する。

命題２の仮定より、自サーバ１００ｉのサーバ優先度２１０は１である。このとき、補題３より、相手サーバのサーバ優先度は１ではない。このとき補題１より、自サーバ１００ｉは必ず最後に処理を行っていたサーバである。

このように、補題１，２，３が成り立てば、命題１及び２は成り立つ。

【０１２０】

次に、補題１，２，３それぞれについて、仮定が成り立てば結論が成り立つことの証明を行い、最後に、補題の仮定が成り立つことを説明する。

【０１２１】

補題１の仮定から、結論が成り立つことの証明：

自サーバ１００ｉのサーバ優先度２１０に１がセットされたということは、その時点で、自サーバ１００ｉはシングルマスタ状態であったことを意味する。その後、相手サーバ１００ｊは１度もマスタあるいはシングルマスタになっていないということは、相手サーバ１００ｊはその時点から１度も処理を行っていないということである。よって、最後に処理を行ったのは、自サーバ１００ｉである。

【０１２２】

自サーバ１００ｉのサーバ優先度２１０に１がセットされた時点より後に、当該優先度２１０が変更されていないということは、自サーバの現在の優先度２１０は１である。

仮定から、現在の相手サーバ１００ｊのサーバ優先度２１０は１ではない。

よって、自サーバ１００ｉのサーバ優先度２１０が１であり、相手サーバ１００ｊのサーバ優先度２１０が１ではないとき、最後に処理を行ったのは自サーバ１００ｉである。

【０１２３】

補題２の仮定から、結論が成り立つことの証明：

自サーバ１００ｉのサーバ優先度２１０に２がセットされたということは、その時点で、自サーバ１００ｉはマスタ状態であったことを意味する。マスタ状態になるためには、相手サーバ１００ｊは必ずスレーブ状態である必要があり、したがって、その時点での相手サーバ１００ｊのサーバ優先度２１０は３である。

【０１２４】

その後、相手サーバ１００ｉは１度もマスタあるいはシングルマスタになっていないということは、相手サーバ１００ｊはその時点から１度も処理を行っていないということである。よって、最後に処理を行ったのは、自サーバ１００ｉである。

【０１２５】

また、相手サーバ１００ｊのサーバ優先度２１０は、マスタ或いはシングルマスタにならなければ変わらないので、相手サーバ１００ｊのサーバ優先度２１０は３のままである。

【０１２６】

また、自サーバ１００ｉのサーバ優先度２１０に２がセットされた時点より後に当該優先度２１０が変更されていないということは、自サーバ１００ｉの現在の優先度２１０は２である。

【０１２７】

よって、自サーバ１００ｉのサーバ優先度２１０が２であり、相手サーバ１００ｊのサーバ優先度２１０が３であるとき、最後に処理を行ったのは自サーバ１００ｉである。

【０１２８】

補題３の仮定から、結論が成り立つことの証明：

両サーバ１００ｉ，１００ｊが停止した状態のとき、サーバ優先度２１０は常に互いに異なる状態をとることを証明するには、サーバ優先度２１０が常に互いに異なる状態をとることを証明すればよい。そのためには、次の４つの仮定３－１～３－４を証明すればよい。

【０１２９】

仮定３－１：

サーバ優先度２１０の初期値は互いに異なっている。

仮定３－２

片方のサーバのサーバ優先度２１０が１のとき、もう片方のサーバのサーバ優先度２１０は１にならない。

仮定３－３

片方のサーバのサーバ優先度２１０が２のとき、もう片方のサーバのサーバ優先度２１０は２にならない。

仮定３－４

片方のサーバのサーバ優先度２１０が３のとき、もう片方のサーバのサーバ優先度２１０は３にならない。

これらが証明できれば、片方のサーバのサーバ優先度２１０が何であっても、もう片方のサーバのサーバ優先度２１０は同じにならないことがいえる。

【０１３０】

仮定３－１は、サーバ優先度２１０の動作より明らかである。

以下で、仮定３－２～３－４を証明する。

仮定３－２の証明：

片方のサーバ（サーバ１００ｐとする）のサーバ優先度２１０が１になったとき、もう片方のサーバ（サーバ１００ｑとする）は停止状態である。この状態で、サーバ１００ｐのサーバ優先度２１０を変えずにサーバ１００ｑのサーバ優先度２１０を１にするには、サーバ１００ｑをシングルマスタ状態にしなければならない。そのためには、両サーバが一旦、停止状態に遷移し、更にサーバ１００ｐがスレーブ状態、サーバ１００ｑが停止状態という状態に遷移しなければならない。しかし、この時点で、サーバ１００ｐのサーバ優先度２１０は１であり、これは両サーバが停止した状態から、マスタースレーブ状態への状態遷移の決定方法に矛盾する。よって、自サーバのサーバ優先度２１０を変更せずに相手サーバがシングルマスタ状態になることは不可能である。つまり、片方のサーバ１００ｐのサーバ優先度２１０が１のとき、もう片方のサーバ１００ｑのサーバ優先

度２１０は１にならない。

【０１３１】

仮定３－３の証明：

片方のサーバ１００ｐのサーバ優先度２１０が２になったとき、もう片方のサーバ１００ｑは必ず３になる。よって、片方のサーバ１００ｐのサーバ優先度が２のとき、もう片方のサーバ１００ｑのサーバ優先度は２にならない。

【０１３２】

仮定３－４の証明：

片方のサーバ１００ｐのサーバ優先度２１０が３になったとき、もう片方のサーバ１００ｑはマスタ状態である。この状態で、サーバ１００ｐのサーバ優先度２１０を変えずにサーバ１００ｑのサーバ優先度２１０を３にするには、サーバ１００ｑをスレーブ状態にしなければならない。そのためには、両サーバ１００ｐ，１００ｑが一旦停止状態に遷移し、さらにサーバ１００ｐが停止状態、サーバ１００ｑがスレーブ状態という状態に遷移しなければならない。しかし、このときの両者のサーバ優先度２１０は３と２であり、サーバ１００ｑの方が優先度２１０が高い。これは両サーバ１００，１００ｊが停止した状態から、マスタースレーブ状態への状態遷移の決定方法に矛盾する。よって、自サーバのサーバ優先度２１０を変更せずに相手サーバがスレーブ状態になることは不可能である。つまり、片方のサーバ１００ｐのサーバ優先度２１０が３のとき、もう片方のサーバ１００ｑのサーバ優先度２１０は３にならない。

このように補題１，２，３について、仮定から、結論が成り立つことが証明できた。

【０１３３】

次に補題１，２の仮定が成り立つことを説明する。

補題１の仮定が成り立つことの証明：

自サーバ１００ｉのサーバ優先度２１０に１がセットされたとき、相手サーバ１００ｊのサーバ優先度２１０が１でなければ、自サーバ１００ｉのサーバ優先度２１０を変えずに、相手サーバ１００ｊがマスタ状態またはシングルマスタ状態になることができないことを、図２の状態遷移図８００を参照して説明する。

【０１３４】

自サーバ１００ｉのサーバ優先度２１０が１にセットされた後に、相手サーバ１００ｊがマスタ状態になるためには、自サーバ１００ｉはスレーブ状態になる必要があり、そのとき自サーバ１００ｉのサーバ優先度２１０は３に変更されてしまう。つまり、自サーバ１００ｉのサーバ優先度２１０を変更せずに相手サーバ１００ｊがマスタ状態になることは不可能である。よって、自サーバ１００ｉのサーバ優先度２１０が１にセットされた後、相手サーバ１００ｊはマスタ状態になっていない。

【０１３５】

自サーバ１００ｉのサーバ優先度２１０が１にセットされたより後に、相手サーバ１００ｊがシングルマスタになるためには、図２の状態遷移図８００から、一旦、相手サーバ１００ｊがマスタ、自サーバ１００ｉがスレーブ、もしくはその逆に、相手サーバ１００ｊがスレーブ、自サーバ１００ｉがマスタという状態になるか、或いは、一旦両サーバ１００ｉ，１００ｊ共に停止状態になって、その後に相手サーバ１００ｊがシングルマスタにならなければならない。

【０１３６】

ところが、一方がマスタで、他方がスレーブになる場合は、自サーバ１００ｉのサーバ優先度２１０が変わってしまうので不可能である。また、両サーバ共に停止状態になり、相手サーバ１００ｊがシングルマスタになるためには、判定方法の定義から、相手サーバ１００ｊのサーバ優先度２１０が１でなければならない。これは仮定に反する。よって、自サーバ１００ｉのサーバ優先度２１０に１がセットされた後、相手サーバ１００ｊはシングルマスタ状態になっていないといえる。

以上のことから、自サーバ１００ｉのサーバ優先度２１０を変えずに、相手サーバ１００ｊがマスタ状態またはシングルマスタ状態になることはできない。

【０１３７】

補題２の仮定が成り立つことの証明：

自サーバ１００ｉのサーバ優先度２１０が２にセットされたとき、自サーバ１００ｉのサーバ優先度２１０を変えずに、相手サーバ１００ｊがマスタ状態また

はシングルマスタ状態になることができないことを、図２の状態遷移図８００を参照して説明する。

【０１３８】

自サーバ１００ｉのサーバ優先度２１０が２にセットされた時点での、自サーバ１００ｉの状態は必ずマスタ状態であり、相手サーバ１００ｊの状態は必ずスレーブ状態である。

【０１３９】

自サーバ１００ｉのサーバ優先度２１０が２にセットされた後に、相手サーバ１００ｊがマスタ状態になるためには、自サーバ１００ｉはスレーブ状態になる必要があり、そのとき自サーバ１００ｉのサーバ優先度２１０は３に変更されてしまう。つまり、自サーバ１００ｉのサーバ優先度２１０を変更せずに相手サーバ１００ｊがマスタ状態になることは不可能である。よって、自サーバ１００ｉのサーバ優先度２１０が２にセットされた後、相手サーバ１００ｊはマスタ状態になっていない。

【０１４０】

自サーバ１００ｉのサーバ優先度２１０に２がセットされた後に、自サーバ１００ｉのサーバ優先度２１０を変えることなく相手サーバ１００ｊがシングルマスタ状態になることは可能である。しかし、その後、自サーバ１００ｉのサーバ優先度２１０を変えることなく、相手サーバ１００ｊのサーバ優先度２１０を３にするためには、図２の状態遷移図８００から、両サーバ１００ｉ，１００ｊが一旦停止状態に遷移し、更に自サーバ１００ｉが停止状態、相手サーバ１００ｊがスレーブ状態という状態に遷移しなければならない。ところが、この時点での両者のサーバ優先度２１０は自サーバ１００ｉが２で相手サーバ１００ｊが１であり、これは両サーバ１００ｉ，１００ｊが停止した状態から、マスタースレーブ状態への状態遷移の決定方法に矛盾する。よって、自サーバ１００ｉのサーバ優先度２１０を変更せずに、相手サーバ１００ｊがシングルマスタ状態になることは不可能である。つまり、自サーバ１００ｉのサーバ優先度２１０が２にセットされた後、相手サーバ１００ｊはシングルマスタ状態になっていない。

【０１４１】

以上により、自サーバ１００ｉのサーバ優先度２１０が２にセットされたより後に、相手サーバ１００ｊはマスタまたはシングルマスタになっていない。

よって、両サーバ１００ｉ，１００ｊが停止した状態のとき、どちらかのサーバのサーバ優先度２１０が２で、もう一方のサーバ優先度２１０が３であった場合、サーバ優先度２１０が２のサーバ、つまり優先度が高い方のサーバが最後に処理をしていたサーバであるといえる。

【０１４２】

次に、強制開始を行う場合の動作を説明する。

図７に、強制開始がある場合の状態遷移及びサーバ優先度２１０の遷移を示す。

図７に示す強制開始がある場合の遷移が、図６に示した強制開始がない場合と相違する点は、（Ｘ　Ｘ）の状態からの２－１または２－２の遷移（図７において太線の矢印で示す遷移）で、サーバ優先度２１０の低い方のサーバを、強制的にシングルマスタにする遷移があるということである。

【０１４３】

そのため、（Ｘ　Ｘ），（Ｘ　ＳＭ），（ＳＭ　Ｘ）の３状態のとき、［１　１］というサーバ優先度２１０を持つ場合がある（図において※印が付されている優先度）。

【０１４４】

例えば、（Ｘ　Ｘ）の状態で、サーバ優先度が［１　３］や、［１　２］のとき、（Ｘ　ＳＭ）へ強制開始によって状態遷移を行うことができる。そのとき、サーバ優先度は［１　１］になる。この状態でシングルマスタ状態のサーバに障害が発生した場合、（Ｘ　Ｘ）で［１　１］になる。

【０１４５】

このように、強制開始の結果、両サーバのサーバ優先度２１０が等しい状態が発生した場合、優先度の高いサーバを判定することができなくなる。したがって、両サーバが障害から復帰した場合に、最後に処理を行っていたサーバを判定することができなくなる。この場合は、もう１度強制開始を行う必要がある。

【０１４６】

また、強制開始の結果、上記の如く（Ｘ　Ｘ）のときに［１　１］という状態があるため、一方のサーバが障害から復帰しても、２－１または２－２の遷移を自動的に行うことができなくなる。２－１または２－２の遷移をさせるためには、必ずもう１度強制開始を行う必要がある。

【０１４７】

以上に述べたように、本実施形態においては、サーバの状態を、処理を行い、且つ行っている処理を引き継ぐ相手が存在する「マスタ」と、処理は行うが、行っている処理を引き継ぐ相手が存在しない「シングルマスタ」と、処理を行っていないが、引き継ぎのための情報を受け取っている「スレーブ」と、処理を行っておらず、且つ引き継ぎのための情報を受け取っていない「停止」の４つの状態に分類して作成された状態遷移図８００に従って、サーバ１００ａ，１００ｂの状態遷移を行うと共に、この状態の遷移で決まるサーバ優先度２１０という状態変数をローカルなディスク装置２００ａ，２００ｂに記録し、当該状態遷移と状態変数（サーバ優先度２１０）を用いて、最後まで処理を行っていたサーバを判定するようにしたので、次のような効果を得ることができる。

まず、強制開始を使わない場合、サーバ１００ａ．１００ｂの片方または両方（つまり少なくとも一方のサーバ）に、どのようなタイミングで障害発生、或いは障害復帰が起こっても、サーバ１００ａ，１００ｂに障害が発生し、復帰したときに、当該サーバ自身が処理を行うべきサーバであるか否かを判定することができる。

【０１４８】

特に本実施形態においては、前記した自動運転を行うための動作（１）～（５）を、２台のサーバ１００ａ，１００ｂ内で状態遷移のできるクラスタ管理部１１０を使うことで実現し、動作（１）（２）で必要なサーバの判定を、状態遷移の度にサーバ優先度処理部７００内の状態書き込み部７１０がローカルなディスク装置に遷移状態に対応したサーバ優先度２１０を記録して、両サーバ１００ａ，１００ｂのサーバ優先度２１０をサーバ優先度処理部７００内の比較処理部７２０で比較することによって実現しているため、共有ディスク装置や時刻情報を使うことなく、自動運転を実現できる。

【０１４９】

また、強制開始を使う場合、動作（２）において、復帰したサーバが処理を引き継ぐサーバであると判定できない場合でも、必要に応じて強制的に処理を引き継ぐことができる。この場合においても、両サーバが（Ｘ　Ｘ）で［１　１］という特別な状態に遷移した場合を除き、自動的に処理を継続することができる。また、上記特別な状態の場合も、再度強制開始を用いて処理を継続することができる。

【０１５０】

【発明の効果】

以上詳述したように本発明によれば、所定の状態遷移図に従う状態の遷移で決まるサーバ優先度という状態変数をローカルな記憶装置に記録し、当該状態遷移と状態変数を用いて、最後まで処理を行っていたサーバを判定するようにしたので、処理引き継ぎの回数の制限をなくして自動運転を実現すると共に、判定精度を向上することができる。

【図面の簡単な説明】

【図１】

本発明の一実施形態に係る高可用性計算機システムの構成を示すブロック図。

【図２】

同実施形態で適用される状態遷移図８００を示す図。

【図３】

図１中の状態書き込み部７１０によるサーバ優先順位変更処理を説明するためのフローチャート。

【図４】

図１中の比較処理部７２０によるサーバ優先度判定処理を説明するためのフローチャート。

【図５】

図１中のクラスタ管理部１１０による障害復帰時状態遷移処理を説明するためのフローチャート。

【図６】

各サーバがローカルに持つサーバ優先度を、強制開始がない場合について、図２の状態遷移図８００と対応付けて示す図。

【図７】

各サーバがローカルに持つサーバ優先度を、強制開始がある場合について、図２の状態遷移図８００と対応付けて示す図。

【符号の説明】

１００ａ，１００ｂ…サーバ

１１０…クラスタ管理部（状態遷移手段、最終処理計算機判定手段）

２００ａ，２００ｂ…ディスク装置（記憶装置）

２１０…サーバ優先度

３００…コマンド送信用計算機

４００…処理・引き継ぎ部

４１０…処理制御部

４２０…引き継ぎ制御部

５００…ネットワーク

７００…サーバ優先度処理部

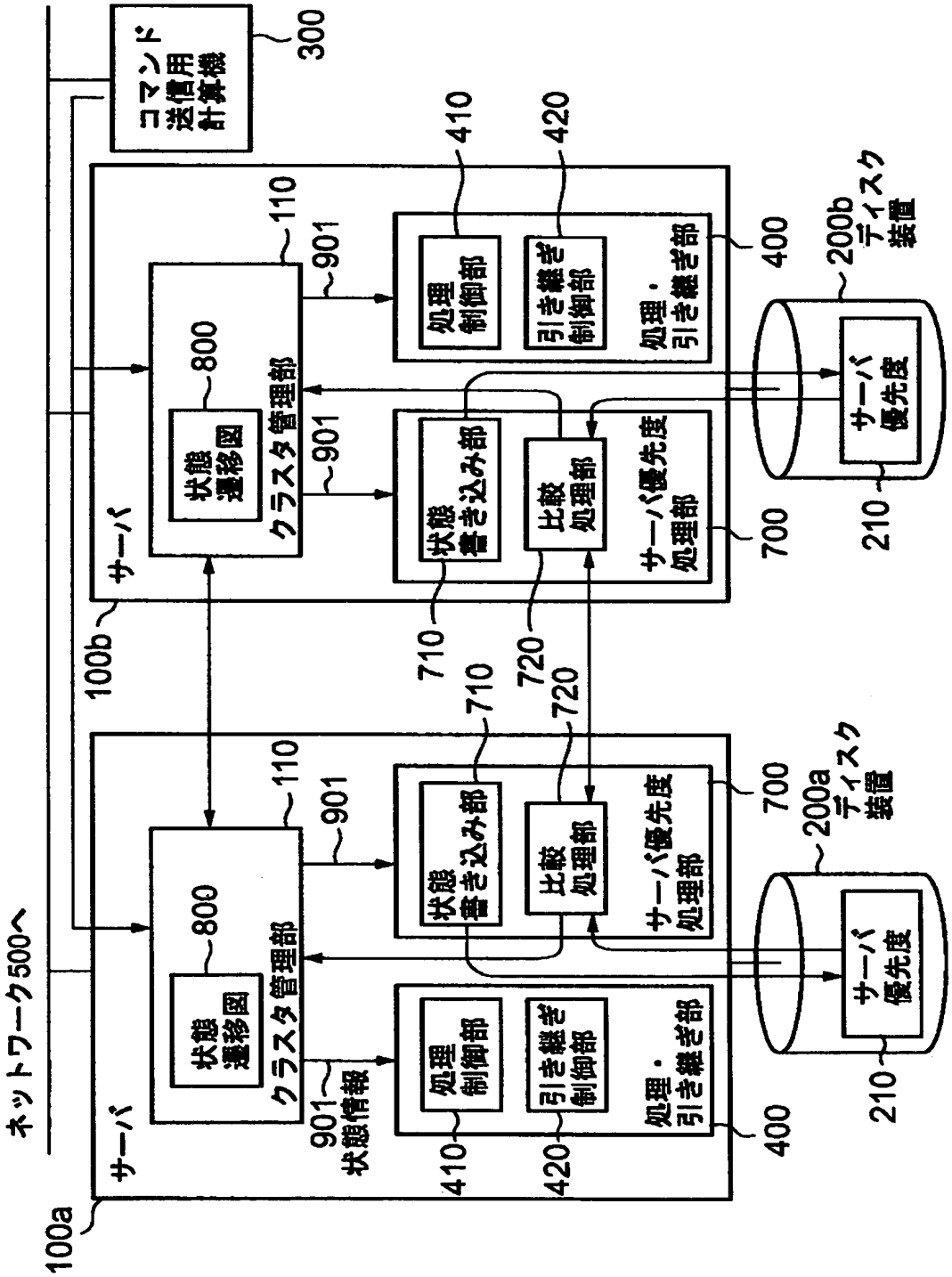
７１０…状態書き込み部

７２０…比較処理部（優先度判定手段）

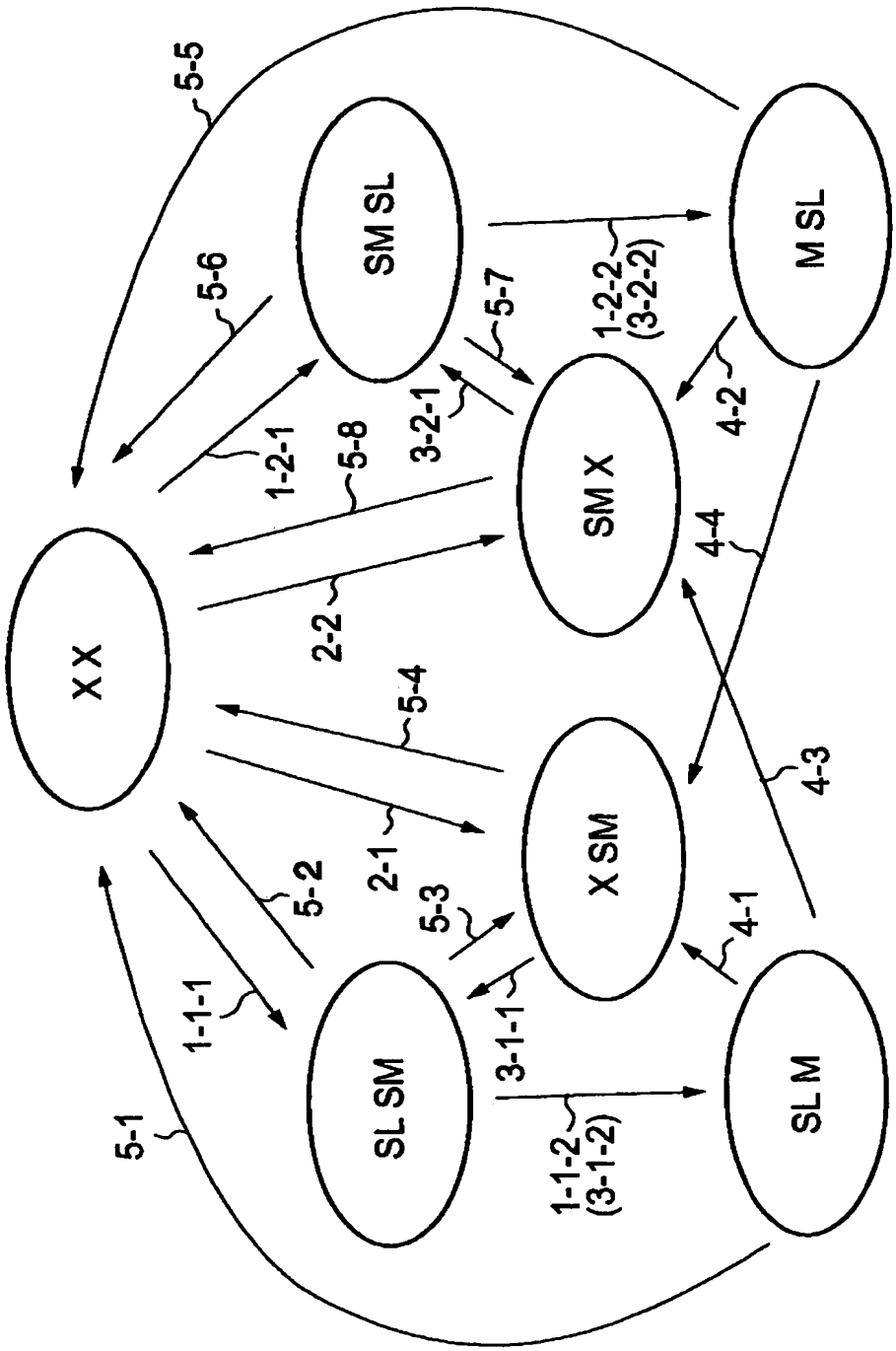
９０１…状態変化情報

【書類名】　図面

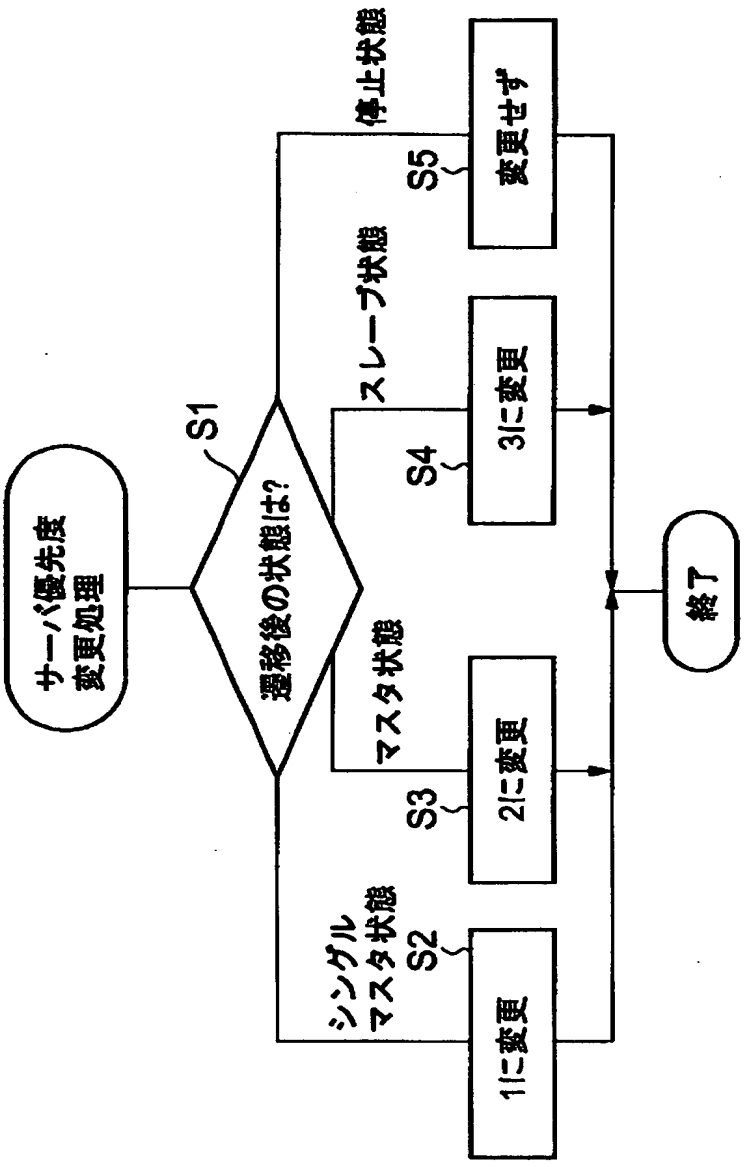
【図１】


ネットワーク500へ
コマンド送信用計算機
300
サーバ
100a
100b
クラスタ管理部
状態遷移図
800
110
901
状態情報
処理制御部
引き継ぎ制御部
処理・引き継ぎ部
400
410
420
状態書き込み部
比較処理部
サーバ優先度処理部
700
710
720
ディスク装置
200a
200b
サーバ優先度
210

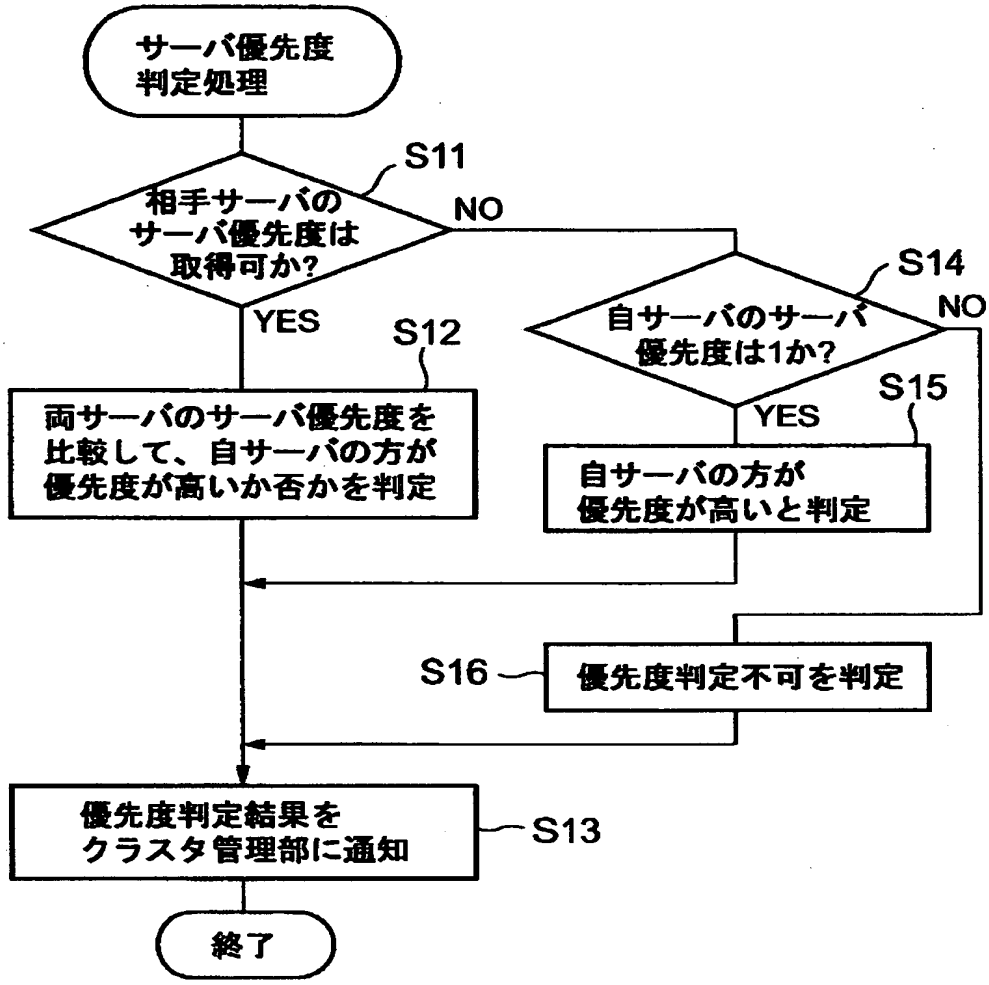
【図２】


X X
SM SL
M SL
SM X
X SM
SL SM
SL M
5-1
5-2
5-3
5-4
5-5
5-6
5-7
5-8
1-1-1
1-1-2
(3-1-2)
1-2-1
1-2-2
(3-2-2)
2-1
2-2
3-1-1
3-2-1
4-1
4-2
4-3
4-4

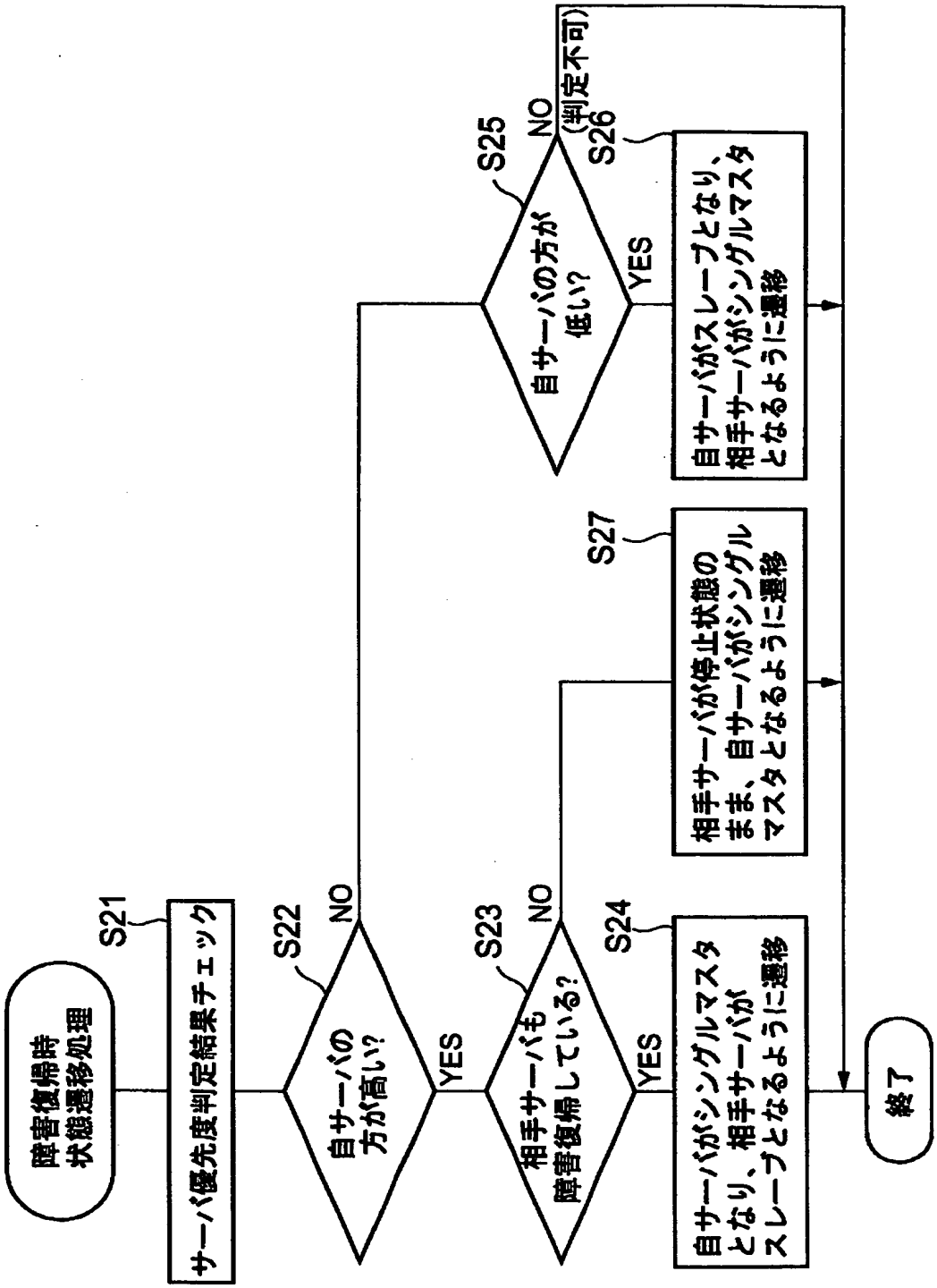
【図３】


サーバ優先度
変更処理
S1
遷移後の状態は?
停止状態
S5
変更せず
スレーブ状態
S4
3に変更
マスタ状態
S3
2に変更
シングル
マスタ状態
S2
1に変更
終了

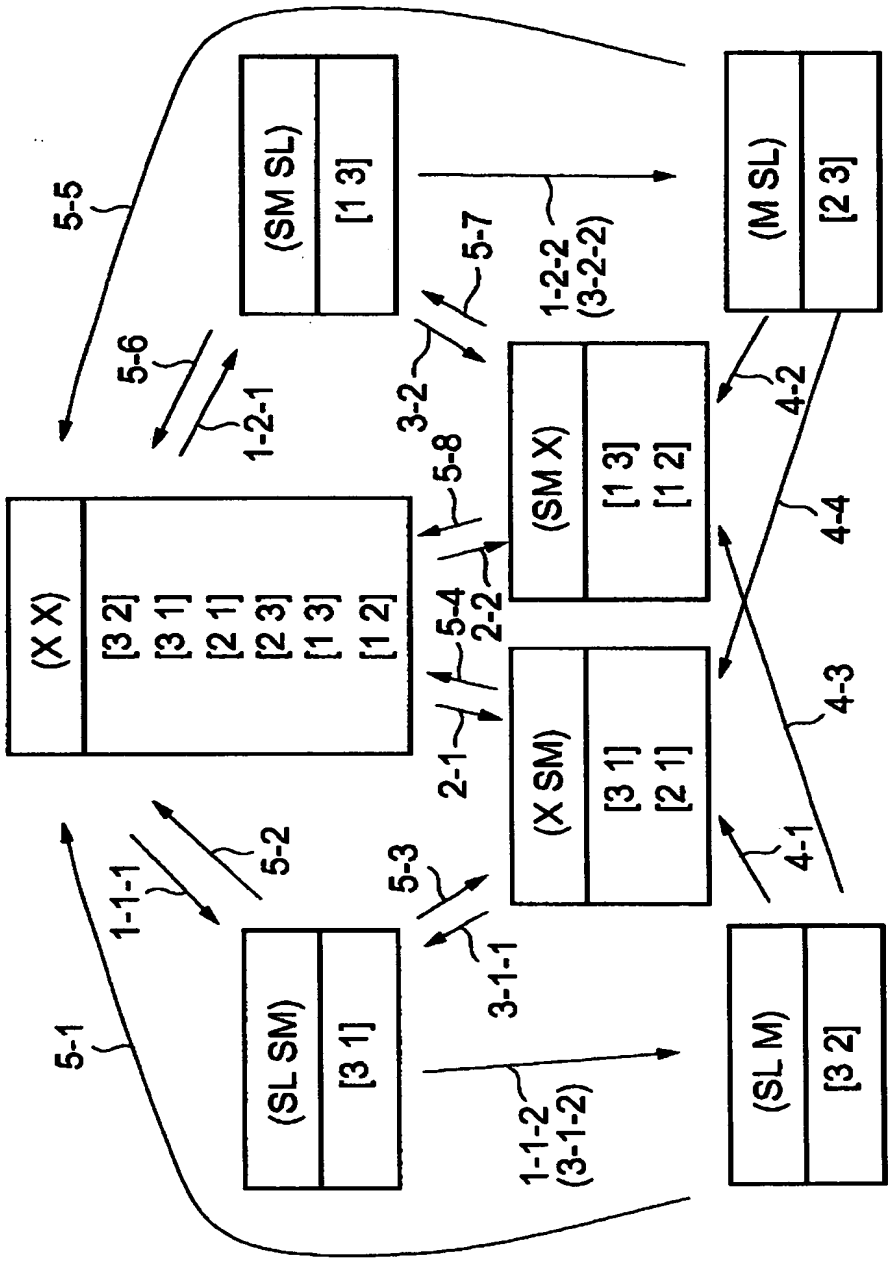
【図４】


サーバ優先度
判定処理
S11
相手サーバの
サーバ優先度は
取得可か?
NO
YES
S12
両サーバのサーバ優先度を
比較して、自サーバの方が
優先度が高いか否かを判定
S14
自サーバのサーバ
優先度は1か?
NO
YES
S15
自サーバの方が
優先度が高いと判定
S16
優先度判定不可を判定
優先度判定結果を
クラスタ管理部に通知
S13
終了

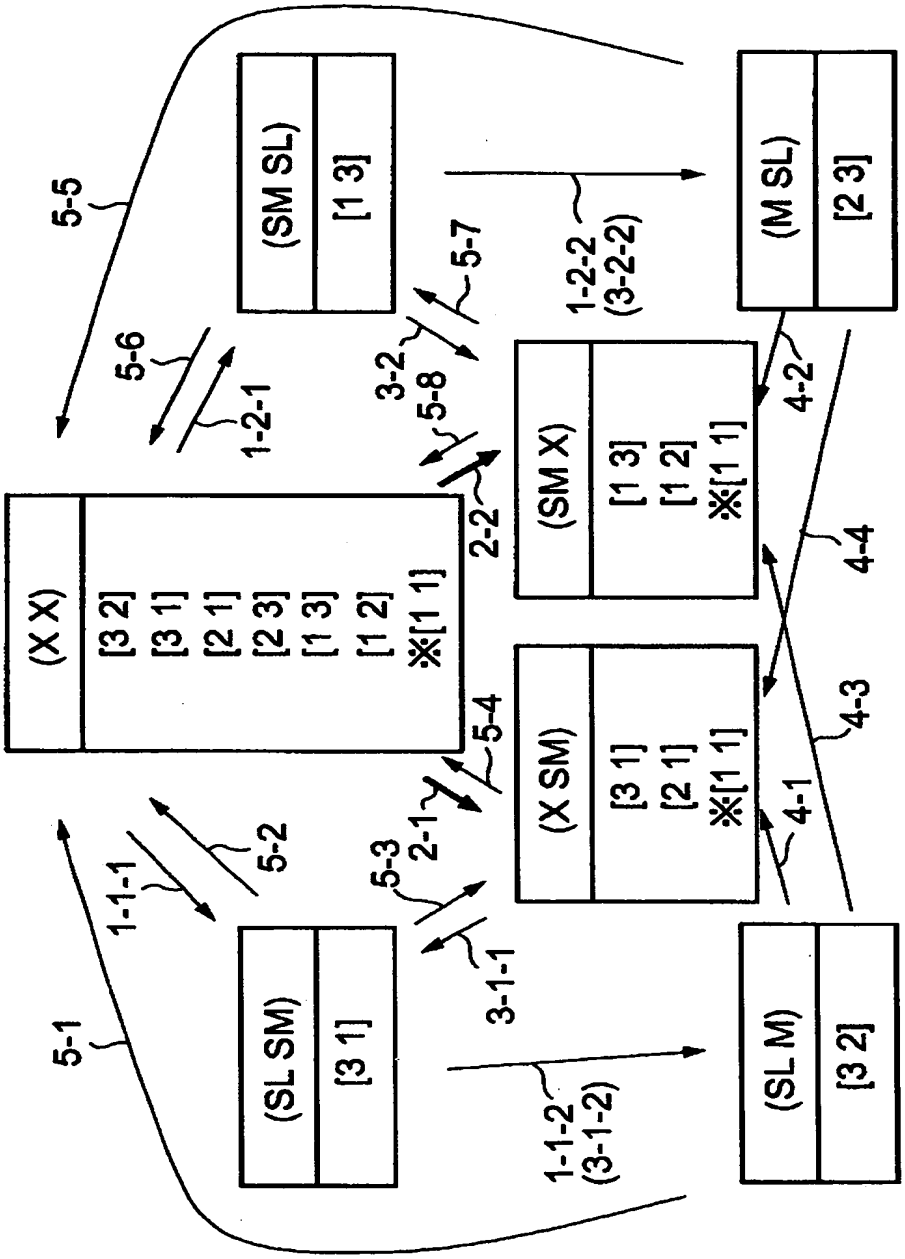
【図５】


障害復帰時
状態遷移処理
S21
サーバ優先度判定結果チェック
S22
自サーバの
方が高い?
NO
YES
S23
相手サーバも
障害復帰している?
NO
YES
S24
自サーバがシングルマスタ
となり、相手サーバが
スレーブとなるように遷移
S25
自サーバの方が
低い?
NO
(判定不可)
YES
S26
自サーバがスレーブとなり、
相手サーバがシングルマスタ
となるように遷移
S27
相手サーバが停止状態の
まま、自サーバがシングル
マスタとなるように遷移
終了

【図６】


(X X)
[3 2]
[3 1]
[2 1]
[2 3]
[1 3]
[1 2]
(SM SL)
[1 3]
(M SL)
[2 3]
(SM X)
[1 3]
[1 2]
(X SM)
[3 1]
[2 1]
(SL SM)
[3 1]
(SL M)
[3 2]
5-1
5-2
5-3
5-4
5-5
5-6
5-7
5-8
1-1-1
1-1-2
(3-1-2)
1-2-1
1-2-2
(3-2-2)
2-1
2-2
3-1-1
3-2
4-1
4-2
4-3
4-4

【図７】


(X X)
[3 2]
[3 1]
[2 1]
[2 3]
[1 3]
[1 2]
※[1 1]
(SM SL)
[1 3]
(M SL)
[2 3]
(SM X)
[1 3]
[1 2]
※[1 1]
(X SM)
[3 1]
[2 1]
※[1 1]
(SL SM)
[3 1]
(SL M)
[3 2]
5-1
5-2
5-3
5-4
5-5
5-6
5-7
5-8
1-1-1
1-1-2
(3-1-2)
1-2-1
1-2-2
(3-2-2)
2-1
2-2
3-1-1
3-2
4-1
4-2
4-3
4-4

【書類名】　　要約書

【要約】

【課題】ローカルな情報だけを使って最後に処理を行っていたサーバを判定することにより、処理引き継ぎの回数の制限をなくして自動運転を実現すると共に、判定精度の向上が図れるようにする。

【解決手段】サーバ１００ａ，１００ｂのクラスタ管理部１１０は、少なくとも一方のサーバで障害が発生した場合と障害から復帰した場合、状態遷移図８００に従うサーバ状態の遷移を行う。状態書き込み部７１０は、状態変化情報９０１の示すサーバ状態で決まるサーバ優先度２１０をディスク装置２００ａ，２００ｂに記録する。比較処理部７２０は、サーバ１００ａ，１００ｂに障害が発生し、その後少なくとも自サーバが障害復帰した際、少なくとも自サーバの優先度２１０に基づき自サーバの方が優先度が高いかを判定する。クラスタ管理部１１０は、この判定結果と状態遷移図８００に基づき、対応する状態遷移を行う。

【選択図】　　図１

# 出　願　人　履　歴　情　報

識別番号　　〔０００００３０７８〕

１．変更年月日　　１９９０年　８月２２日
　［変更理由］　　新規登録
　　住　所　　神奈川県川崎市幸区堀川町７２番地
　　氏　名　　株式会社東芝